

SeeBeyond ICAN Suite

SeeBeyond ICAN Suite Deployment Guide

Release 5.0.5



The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

SeeBeyond, e*Gate, e*Way, and e*Xchange are the registered trademarks of SeeBeyond Technology Corporation in the United States and/or select foreign countries. The SeeBeyond logo, SeeBeyond Integrated Composite Application Network Suite, eGate, eWay, eInsight, eVision, eXchange, eView, eIndex, eTL, ePortal, eBAM, and e*Insight are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2004 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.

Version 20041118175618.

Contents

List of Figures	8
-----------------	---

List of Tables	9
----------------	---

Chapter 1

Introduction	10
---------------------	-----------

About This Document	10
What's in This Document	10
Scope	11
Intended Audience	11
Document Conventions	11
Screenshots	12
Related Documents	12
SeeBeyond Web Site	13
SeeBeyond Documentation Feedback	13

Chapter 2

Overview of the ICAN Suite	14
-----------------------------------	-----------

eGate Integrator	15
Integration Model	15
User Interfaces	15
Enterprise Designer	15
Enterprise Manager	15
System Architecture	16
Projects	16
eInsight Business Process Manager	16
eVision Studio	17
ePortal Composer	18
eTL Integrator	18
eXchange Integrator	18
eView Studio	19

eIndex Global Identifier Composite Application	19
eBAM Studio	19

Chapter 3

Requirements Analysis and Deployment Planning	20
Introduction	20
Requirements Analysis	22
System-Specific Needs	22
Operation and Performance Needs	23
Personnel and Training Needs	24
Business Planning Needs	24
Deployment Planning	26
Determining Overall Objectives	26
Initiation Steps	27
Hold a Project Kick-off Meeting	27
Ensure That Hardware and Software Are Ready	27
Complete Installation Test, Installation, and Checklist	27
Establish the Change Management Process	28
Creating the Planning Documents	28
Deployment Project Plan	28
Functional Requirements Specification	29
Technical Requirements Specification	30
Initial Test Plan	30

Chapter 4

System Design and Development	32
Introduction	32
Service-Oriented Architecture	34
Overview	34
Architecture Layers	35
Existing Systems	35
Elemental Business Services	35
Composed Business Services	36
Composite Applications	36
Invoking Services	37
System Design	38
Identifying External Systems	38
Configuring ICAN Suite Components	38
Hardware and Network Connections	38
System Development	39
Creating Users	39
Multiple Repositories	39
Naming Conventions	40
Prefixes	40

Suffixes	41
Project Teams with Multiple Developers	42
Subprojects and Multiple Projects	42
Version Control and Access Control Lists	42
Activating Multiple Deployment Profiles	42
Logical Hosts	43
Error Handling	44
Variables and Constants	44
Multiple Environments	44
Creating Multiple Run-Time Components	45
Directory Structure for Multiple Logical Hosts	45
Optimizing Your System	46
Project Size	46

Chapter 5

Testing, Transition to Production, and Maintenance	47
Introduction	47
Pre-Transition Testing	49
Testing Methodology	50
Test Plan	50
Type of Data To Use	50
Testing the Output	50
Responsibility for Testing	51
Unit Testing	51
Integration Testing	51
Partial Integration Testing	51
Complete System Testing	52
Performance Testing	52
Acceptance Testing	52
Troubleshooting	53
ICAN Monitor	53
Log Files	53
Alert Agent	53
SNMP Agent	53
Transition to Production	54
Post-Transition Maintenance	54
Monitoring System Activity	54
ICAN Monitor	54
Log Files	55
Repository Backup	55
Implementing Changes	55
Summary	56

Chapter 6

Determining System Requirements	57
Introduction	57
Initial Considerations	58
Estimating Requirements	58
Factors	58
Guidelines	59
General Considerations	60
Minimum System Requirements	61

Chapter 7

Configuring Failover Support in a Windows Clustering Environment	63
Windows Clustering Overview	63
Configuring Support for Repository Failover	64
Overview	64
Requirements	64
Creating the ICAN Group	65
Creating a Physical Disk Resource	65
Creating an IP Address Resource	67
Creating a Network Name Resource	68
Installing the Repository as a Windows Service	69
Duplicating the Registry Keys	69
Creating a Generic Service Resource	70
Using the Repository in a Windows Clustering Environment	72
Configuring Support for Logical Host Failover	73
Overview	73
Requirements	73
Creating the ICAN Group	74
Creating a Physical Disk Resource	74
Creating an IP Address Resource	75
Creating a Network Name Resource	75
Installing the Logical Host as a Windows Service	76
Duplicating the Registry Keys	76
Creating a Generic Service Resource	77
Using the Logical Host in a Windows Clustering Environment	78

Chapter 8

Case Study	79
Requirements Analysis and Deployment Planning	79
System Design and Development	80

Testing, Transition to Production, and Maintenance	82
--	----

Appendix A

eGate Integrator on HP NonStop Server	83
Introduction to eGate on HP NonStop Server	83
Installation and Configuration of eGate Integrator on HP NonStop Server	84
LHInit Program	84
Repository and Pathway	85
Repository Under Pathway for HP NonStop Server	85
Transaction Monitoring Facility	86
Configuring for Maximum Scalability	86
Configuring Logical Hosts to use Parallel Library TCP/IP	87
Troubleshooting	87
Signal 31 Message	87
Repository Start Time Is Slow	88

Glossary	89
-----------------	-----------

Index	97
--------------	-----------

List of Figures

Figure 1	ICAN Suite	14
Figure 2	Business Process Model Example	16
Figure 3	Page Layout Example	17
Figure 4	Deployment Phases	21
Figure 5	Information-Gathering Cycle in Requirements Analysis Phase	25
Figure 6	System Design and Development Phase	33
Figure 7	Connectivity Map with Two Services	34
Figure 8	Architecture Layers	35
Figure 9	Naming Subprojects	41
Figure 10	Sharing Logical Hosts Between Projects	43
Figure 11	Multiple Logical Hosts on a Single Computer (Windows Example)	45
Figure 12	Testing, Transition to Production, and Maintenance Phases	48
Figure 13	Change Management Cycle	49
Figure 14	Cluster With Two Nodes	63
Figure 15	New Resource Dialog Box	66
Figure 16	Possible Owners Dialog Box	66
Figure 17	Dependencies Dialog Box	67
Figure 18	TCP/IP Address Parameters Dialog Box	68
Figure 19	Network Name Parameters Dialog Box	69
Figure 20	Generic Service Parameters Dialog Box	71
Figure 21	Registry Replication Dialog Box	71
Figure 22	Receive Activity in Business Process	80
Figure 23	User-Defined Object Type Definition for Inventory Request	81
Figure 24	Decision Branching Activity in Business Process	81
Figure 25	Catch All Exceptions Element in Business Process	81
Figure 26	Utilizing Multiple CPUs	86

List of Tables

Table 1	Document Conventions	11
Table 2	Deployment Project Plan	28
Table 3	Functional Requirements Specification	29
Table 4	Technical Requirements Specification	30
Table 5	Initial Test Plan	31
Table 6	Web Service Standards	37
Table 7	Suggested Prefixes for Project Components	40
Table 8	Suggested Prefixes for Environment Components	41
Table 9	JMS Client Recommended Setting (Small Project)	46
Table 10	Integration Server Recommended Settings (Small Project)	46
Table 11	Testing Types	49
Table 12	General Considerations	60
Table 13	Windows System Requirements	61
Table 14	Windows Repository and Logical Host System Requirements	61
Table 15	HP Tru64 System Requirements	61
Table 16	HP-UX System Requirements	61
Table 17	IBM AIX System Requirements	61
Table 20	HP NonStop Server System Requirements	62
Table 21	IBM z/OS System Requirements	62
Table 18	Red Hat and SuSE Linux System Requirements	62
Table 19	Sun Solaris System Requirements	62
Table 22	eGate 5.0 Terms	95

Introduction

This chapter introduces you to this guide, its general purpose and scope, its organization, and writing conventions. It also provides sources of related documentation and information.

What's in This Chapter

- [“About This Document” on page 10](#)
- [“Related Documents” on page 12](#)
- [“SeeBeyond Web Site” on page 13](#)
- [“SeeBeyond Documentation Feedback” on page 13](#)

1.1 About This Document

1.1.1 What's in This Document

This document includes the following information:

- **Chapter 1, “Introduction”** introduces you to this guide, its general purpose and scope, its organization, and writing conventions. It also provides sources of related documentation and information.
- **Chapter 2, “Overview of the ICAN Suite”** provides a general overview of the ICAN Suite.
- **Chapter 3, “Requirements Analysis and Deployment Planning”** describes the first two phases of deploying the ICAN Suite: requirements analysis and deployment planning.
- **Chapter 4, “System Design and Development”** describes the third phase of deploying the ICAN Suite: system design and development.
- **Chapter 5, “Testing, Transition to Production, and Maintenance”** describes the final three phases of deploying the ICAN Suite: pre-transition testing, transition to production, and post-transition maintenance.
- **Chapter 6, “Determining System Requirements”** offers guidelines to help you determine the system requirements for the deployment of the ICAN Suite.

- **Chapter 7, “Configuring Failover Support in a Windows Clustering Environment”** describes how to configure failover support for the Repository and the Logical Host using Microsoft Windows 2000 and Windows Server 2003 clustering technologies.
- **Chapter 8, “Case Study”** uses a Web Order scenario to illustrate the various phases of ICAN Suite deployment.
- **Appendix A, “eGate Integrator on HP NonStop Server”** provides information about deploying and maintaining eGate Integrator on an HP NonStop Server system.

In addition, this document includes a **“Glossary” on page 89**.

1.1.2 Scope

The *SeeBeyond ICAN Suite Deployment Guide* provides deployment planning guidelines and deployment strategies for the SeeBeyond® Integrated Composite Application Network Suite™ (ICAN).

The purpose of this guide is to help you successfully complete the following stages of deployment:

- 1 Requirements analysis
- 2 Deployment planning
- 3 System design and development
- 4 Pre-transition testing
- 5 Transition to production
- 6 Post-transition maintenance

1.1.3 Intended Audience

This guide is designed for management, system administrators, and others who are tasked with deployment of the ICAN Suite.

1.1.4 Document Conventions

The following conventions are observed throughout this document.

Table 1 Document Conventions

Text	Convention	Example
Names of buttons, files, icons, parameters, variables, methods, menus, and objects	Bold text	<ul style="list-style-type: none"> ▪ Click OK to save and close. ▪ From the File menu, select Exit. ▪ Select the logicalhost.exe file. ▪ Enter the timeout value. ▪ Use the getClassname() method. ▪ Configure the Inbound File eWay.

Table 1 Document Conventions (Continued)

Text	Convention	Example
Command line arguments, code samples	Fixed font. Variables are shown in <i>bold italic</i> .	bootstrap -p <i>password</i>
Hypertext links	Blue text	See “ Document Conventions ” on page 11
Hypertext links for Web addresses (URLs) or email addresses	Blue underlined text	http://www.seebeyond.com docfeedback@seebeyond.com

1.1.5 Screenshots

Depending on what products you have installed, and how they are configured, the screenshots in this document may differ from what you see on your system.

1.2 Related Documents

For more information about the ICAN Suite, see the following documents:

- *eBAM Studio User’s Guide*
- *eGate Integrator JMS Reference Guide*
- *eGate Integrator User’s Guide*
- *eGate Tutorial*
- *eIndex Global Identifier Composite Application User’s Guide*
- *eInsight Business Process Manager User’s Guide*
- *ePortal Composer User’s Guide*
- *eTL Integrator User’s Guide*
- *eView Studio User’s Guide*
- *eVision Studio User’s Guide*
- *eXchange Integrator User’s Guide*
- *SeeBeyond ICAN Suite Installation Guide*
- *SeeBeyond ICAN Suite Primer*

In addition, each eWay in the ICAN Suite has a user’s guide.

1.3 SeeBeyond Web Site

The SeeBeyond Web site is your best source for up-to-the-minute product news and technical support information. The site's URL is:

<http://www.seebeyond.com>

1.4 SeeBeyond Documentation Feedback

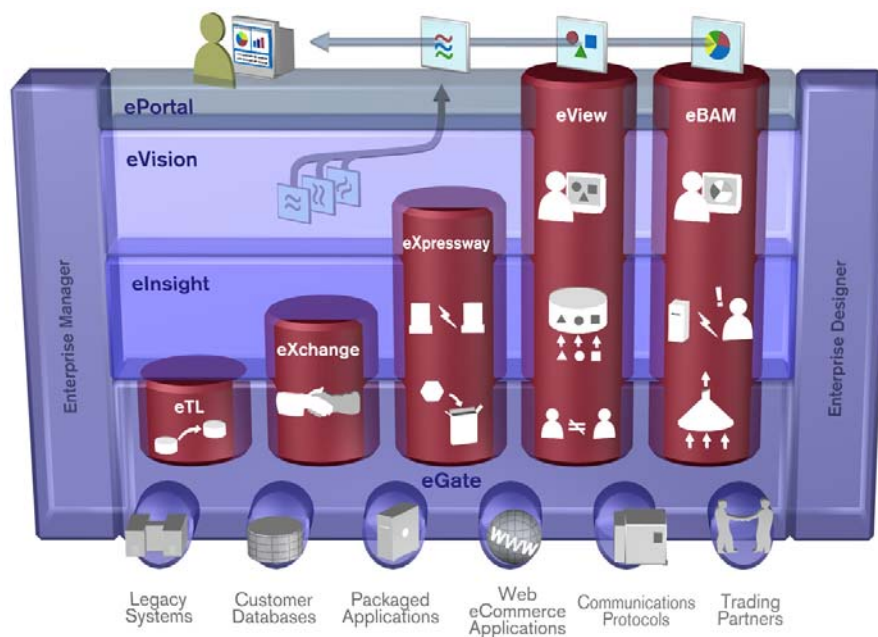
We appreciate your feedback. Please send any comments or suggestions regarding this document to:

docfeedback@seebeyond.com

Overview of the ICAN Suite

This chapter provides a general overview of the ICAN Suite.

Figure 1 ICAN Suite



What's in This Chapter

- “eGate Integrator” on page 15
- “eInsight Business Process Manager” on page 16
- “eVision Studio” on page 17
- “ePortal Composer” on page 18
- “eTL Integrator” on page 18
- “eXchange Integrator” on page 18
- “eView Studio” on page 19
- “eIndex Global Identifier Composite Application” on page 19
- “eBAM Studio” on page 19

2.1 eGate Integrator

eGate Integrator (eGate) is a fully J2EE certified and Web services-based, distributed integration platform that serves as the foundation of the ICAN Suite. eGate provides the core integration platform, comprehensive systems connectivity, guaranteed messaging, and robust transformation capabilities while providing a unified, single sign-on environment for integration development, deployment, monitoring, and management.

The heart of eGate is the Repository, which is a comprehensive store of information common to the entire enterprise.

2.1.1 Integration Model

SeeBeyond addresses application integration by means of an eGate Project, which contains the business logic required to solve the specific problem. The Project contains the various logical components and supporting information required to perform the routing, processing, and caching of messages containing the relevant data from one application to another. All Project information is stored in the Repository.

2.1.2 User Interfaces

The main user interfaces are Enterprise Designer and Enterprise Manager.

Enterprise Designer

Enterprise Designer is used to create and configure the logical components and physical resources of an eGate Project. Through this GUI, you can develop Projects to process and route data through an eGate system. Enterprise Designer is also used by other components of the ICAN Suite.

Enterprise Manager

Enterprise Manager is a Web-based application that works within Microsoft Internet Explorer. It is used throughout the SeeBeyond ICAN Suite for:

- Installing and updating ICAN Suite products
- Accessing ICAN Suite product documentation
- Managing and monitoring run-time components

2.1.3 System Architecture

eGate employs a versatile architecture that is ideally suited to distributed computing environments. As a result, the various components of an eGate system can reside on the same hardware platform (assuming adequate system resources), or be distributed across several different hardware platforms in the enterprise network.

2.1.4 Projects

An eGate Project represents the logical system designed to solve all or part of a business problem. Projects are created using tools contained within Enterprise Designer, such as the Connectivity Map Editor.

2.2 eInsight Business Process Manager

eInsight Business Process Manager (eInsight) delivers business process management features and functions to the ICAN Suite.

Business process management is a strategic orchestration of the movement of information and the flow of complex processes between participants (systems, users, and organizations) to accomplish larger business objectives.

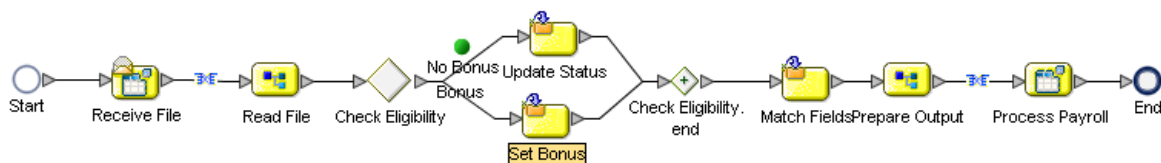
eInsight provides you with a clear view into the internal and external processes of an organization. These processes may be executed by computer systems or employees.

There are two phases of business process management:

- The design phase begins before you start using eInsight and ends once the business process is deployed.
- The run-time phase refers to the tasks that you perform after the business process is deployed.

Figure 2 shows an example of a business process model created with eInsight.

Figure 2 Business Process Model Example



2.3 eVision Studio

eVision Studio is a graphical environment that enables business analysts and Web developers to rapidly create interfaces for composite applications, without the need for advanced programming abilities.

eVision provides two main tools: the Page Layout Designer and the Page Flow Designer. Both tools are integrated within Enterprise Designer.

eVision can be used in a variety of ways as a front end to SeeBeyond ICAN Suite integration solutions. For example:

- In conjunction with eInsight Business Process Manager, eVision allows users to participate in business processes. The users perform workflow tasks with eVision Web pages that are tailored for specific organizational roles.
- eVision can provide the login pages in which users enter their user name and password, which are then authenticated.

Figure 3 shows an example of a Page Layout created with eVision.

Figure 3 Page Layout Example

Tri-Valley
Bank and Loan

Mortgage Loan Approval

Loan Applicant Name:
Social Security Number:
Application Rating:

Property Details	Appraisal Details
Address:	Appraisal Amount: Appraiser: Appraisal Date:

Outstanding Liens

Mechanics:
Judgement:
Tax:
Special Assessment:

Approve Reject

2.4 ePortal Composer

ePortal Composer (ePortal), SeeBeyond's Web portal application program, provides a set of Web-based tools that enable the Web developer/administrator to create Web portals. These portals unify business information and personalize the portal experience based on a user's profile.

A portal allows the organization to present a single, unified view of enterprise data and applications to employees, customers, and partners. Multiple URLs can be assigned to *channels*, which are aggregated within the portal and then collectively presented to users.

Through a Web portlet, called a channel, ePortal enables users to simultaneously view multiple eVision applications, other ICAN-generated user interfaces, and specified Web-enabled enterprise content.

ePortal leverages the ICAN Suite by enabling enterprise-wide access to business processes from a single point of entry, a portal. A portal is a Web site that serves as a gateway for Web-based services and applications.

2.5 eTL Integrator

Extraction Transform and Load (ETL) is a data integration technology that extracts data from several heterogeneous data sources, transforms the data, then loads the data in a uniform format into a target data source.

eTL Integrator is optimized for extracting, transforming and loading bulk data between files and databases and provides an ETL development and run-time environment that is both fully integrated into the ICAN platform and specially optimized for handling very large record sets.

2.6 eXchange Integrator

eXchange Integrator is an application that runs on the SeeBeyond Integration Server in distributed computing environments. eXchange manages interactions with trading partners by facilitating the reception, validation, transmission, and tracking of messages in supported formats.

2.7 eView Studio

eView Studio (eView) provides a flexible framework to allow you to create matching and indexing applications called enterprise-wide master indexes.

eView is an application building tool to help you design, configure, and create a master index that will uniquely identify and cross-reference the business objects stored in your system databases. Business objects can be any type of entity for which you store information, such as customers, members, vendors, businesses, and hardware parts. In eView, you define the data structure of the business objects to be stored and cross-referenced. In addition, you define the logic that determines how data is updated, standardized, weighted, and matched in the master index database.

2.8 eIndex Global Identifier Composite Application

eIndex Global Identifier Composite Application is a health care-oriented, enterprise-wide master person index that creates a single view of person information by maintaining the most current information about the people who participate throughout your organization and by linking information from different locations and computer systems.

2.9 eBAM Studio

BAM (Business Activity Monitoring) involves the collection, aggregation, and presentation of business activity data according to specified Key Performance Indicators (KPIs).

eBAM Studio (eBAM) provides the tools for generating custom, cross-application digital dashboards for defining and monitoring KPIs, which summarize the aggregated business data collected through the eInsight or eGate application layers. The eBAM Web interface allows the business analyst to transform data that has been collected over time into meaningful, rich visual presentations.

eBAM Studio delivers a real-time view of business activities (for example, monitoring service-level agreements for enforcement), allowing the business analyst to identify problems and trends and resolve them proactively.

Notifications are triggered according to preset threshold limits set by the designer. eBAM generates alerts when KPIs fall below or exceed a specified threshold. The alerts can be used to send e-mail messages and/or to launch other processes.

Requirements Analysis and Deployment Planning

This chapter describes the first two phases of deploying the ICAN Suite: requirements analysis and deployment planning.

What's in This Chapter

- [“Introduction” on page 20](#)
- [“Requirements Analysis” on page 22](#)
- [“Deployment Planning” on page 26](#)

3.1 Introduction

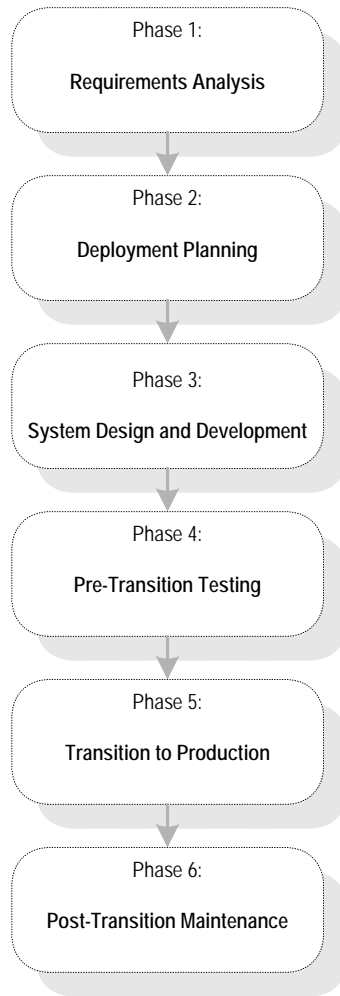
Deploying the ICAN Suite consists of the following phases:

- 1 Requirements analysis
- 2 Deployment planning
- 3 System design and development
- 4 Pre-transition testing
- 5 Transition to production
- 6 Post-transition maintenance

Note: *In a typical deployment, some of the phases overlap with each other. For example, unit and integration testing is usually performed during the system design and development phase.*

Figure 4 shows a diagram of these phases.

Figure 4 Deployment Phases



This chapter explains the first two phases:

- **Requirements analysis:** This guide seeks to give you a road map of how to deploy the ICAN Suite. First, to use a road map, you have to know where you are (analysis) and where you are going (planning). In other words, find out everything you can about your information system setup and business processes. Then, you can decide what information systems and business process needs you want the ICAN Suite to meet.
- **Deployment planning:** Deployment begins when you plan out and schedule how, in view of your analysis and allocated resources, you want to implement your ICAN Suite environment. During this phase, you set up the operation procedure and schedule for the entire deployment project.

The first two phases lay a foundation for the entire deployment project. Poor analysis and planning can cause major problems during the later phases. Thorough, comprehensive analysis and planning can make the deployment project easier, more efficient, and less costly.

3.2 Requirements Analysis

In gathering and analyzing information on your ICAN Suite needs, you must first know what type of information you need. Remember that the ICAN Suite links your current networks, business systems, and applications together into a single, seamless information system. The purpose of this system is to facilitate your current and future business process needs. In as much detail as possible, find out where you are and what you need.

During the requirements analysis phase, you examine your needs and define the properties that the system must possess to meet those needs. Also, you identify system constraints and performance requirements. Define what functions you want the deployed system to perform, but not how the functions work (this task occurs during the system design and development phase).

This section describes the information that you need to gather to facilitate your deployment, by posing a series of relevant questions. The questions are divided into the following categories:

- System-specific
- Operation and performance
- Personnel and training
- Business planning

The examples in this section are general and are only meant to start you thinking in the right direction. You must begin by assembling general information on your needs, categorize that information, and expand on it by filling in necessary details to fully explain each category.

3.2.1 System-Specific Needs

These needs are the basic information systems, network, and database-related requirements that you want the ICAN Suite system to meet. Determine your system-specific needs by asking the following questions:

What existing systems do we need to connect?

Create a complete picture of your current information system setup. Include applications, networks, systems, platforms, and outside information pathways.

Example: An Intel PC LAN with Windows XP network connecting workgroups with office applications, a UNIX system with an Oracle database containing customer information, and UNIX system with IMS tracking financial transactions.

How do we want to do the connecting?

Find out how you want your various systems to talk to each other (communication protocols), which systems must be linked, and the direction of communication.

Example: We have systems A, B, C, and D. Systems A and B use TCP/IP, system C uses SNA, and system D uses SAP. All systems must talk to each other except system D, which needs to communicate only with system A. All communication in all systems is

two-way, except that system C only needs to receive information from the others and not send it.

What are our data requirements?

What types of data do you use, how much, and when?

Example: Our system uses HL7 and X12 data types. On average, our system needs to move about 100,000 messages per day at about 5 MB per message, with 90 percent of the data moving between 8 a.m. and 5 p.m. every Monday through Friday. Peak data loads are generally between 2 p.m. and 4 p.m. on weekdays (60 percent of volume).

What are our system/hardware limitations and constraints?

Installing the ICAN Suite requires that you have the necessary hardware and operating system software and that you purchase (and install if necessary) additional hardware and software to contain the ICAN Suite. Do you want UNIX or Windows? What is your budget for additional hardware and software? Do you have any space limitations in the area where this hardware will reside?

Example: We use Solaris servers for our large-scale systems. We already have Windows client PCs for each of our system developers who will be using Enterprise Designer.

Planning for hardware needs requires special considerations, such as how many systems you need, memory (RAM) required, the number of CPUs you need, and total disk space. [Chapter 6](#) discusses how to analyze and plan for these additional system requirements.

3.2.2 Operation and Performance Needs

Do you have any specific system operation and performance issues? Now is the time to discover, organize, and itemize them by asking the following questions:

What are our performance requirements?

Performance comes down to a trade-off between speed and maintainability. This statement is true overall, as well as being true for the operation of individual system component operations. You must prioritize these needs specifically.

Example: Customer databases must be totally accurate and detailed because the information is used often and is vital to the company. Detailed maintenance of this data is more important than speed of processing. However, our moment-by-moment stock quotations have to be fast and up-to-the minute. Maintainability here is negligible because this data changes so fast that long-term retrieval is not an issue.

What are our internal security requirements?

The ICAN Suite has access security, which allows only certain people to log on to the system and different people to have specific privileges after the log on.

Example: The company allows only five people to log on to the system: one with system administrator privileges, two with operator privileges, and two with monitor privileges.

What are our error-handling and data validation requirements?

How, when, and where in the system does the customer require data to be error checked and validated? Keep in mind that processing speed decreases as checking instances and the detail of error checking increases.

Example: All data passing through our system must be validated to the most thorough extent possible. To facilitate this process, we compiled a complete list of all different data types that require validation.

3.2.3 Personnel and Training Needs

Deploying and maintaining the ICAN Suite solution may require some expanded personnel needs. Consider the following questions:

Do we have personnel trained and able to deploy the system?

The number of resources required to deploy the ICAN Suite solution depends on the size and complexity of the project. You may want to take advantage of SeeBeyond Consulting Services' in-depth product knowledge and experience. In addition, SeeBeyond Education Services offers a comprehensive set of courses.

Example: We need three resources to deploy the solution. The senior member of the team must attend the ICAN Solutions 5.0 class.

Do we have personnel trained and able to maintain the system after deployment?

You must also examine the personnel and training needs of the post-transition maintenance phase.

Example: We need one resource for the maintenance phase.

3.2.4 Business Planning Needs

The ICAN Suite can help you facilitate and improve your overall business processes. Assess your needs in these areas by asking the following questions:

What are our record-keeping and documentation needs?

Make sure that you set up a system for documenting your ICAN Suite operation.

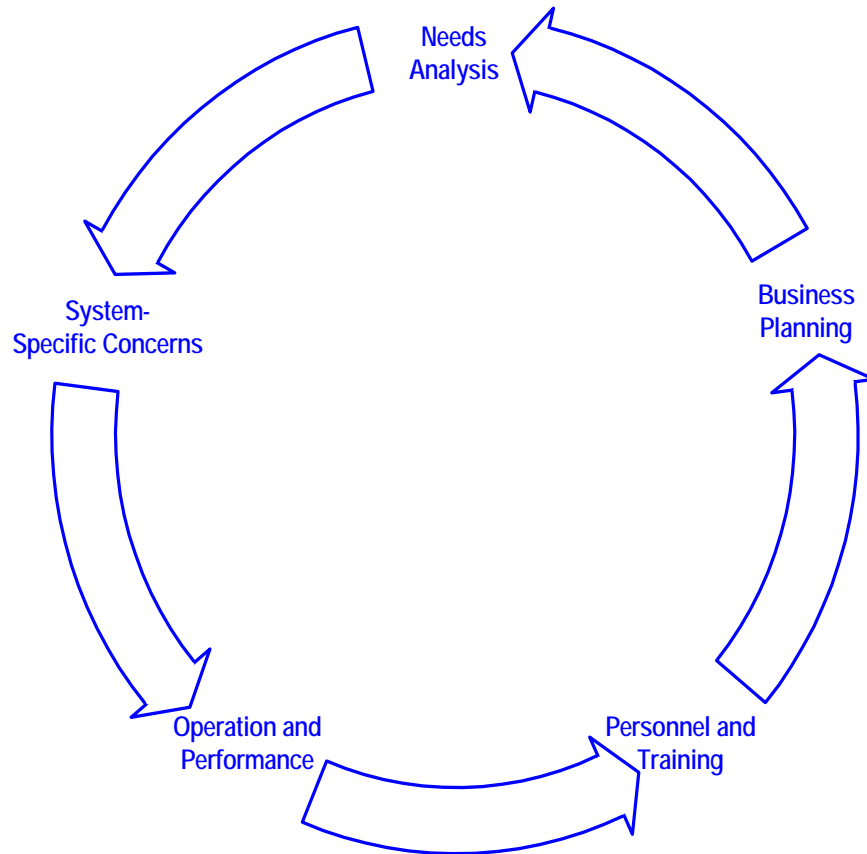
Example: We must put a new methodology in place to document and diagram the total operation of our ICAN Suite operation. In addition, we must keep complete records on that operation.

How do we create a deployment road map?

Plan your deployment well. Choose a deployment project team to carry out the project, and be sure to document your plan in writing. For a small deployment, one person could do this task. Flowcharts and system diagrams are helpful. See **"Deployment Planning" on page 26**.

Figure 5 shows a diagram of the information-gathering cycle in the deployment project's requirements analysis phase.

Figure 5 Information-Gathering Cycle in Requirements Analysis Phase



As you continue the analysis process, allow the results to feed back into your overall analysis. If necessary, repeat the process to fine-tune the information that you have gathered. This approach helps ensure the accuracy and usability of the requirements that you collect.

Once we have the information, what do we do with it?

Complete the process of documenting and organizing your information as correctly and comprehensively as possible. When you finish the requirements analysis phase, you use this information to help you with the next phase: deployment planning. Much of the information will be added to the planning documents.

3.3 Deployment Planning

The deployment planning phase is the second phase in your deployment project. In this phase, you create a road map of what the deployment will look like. You include such criteria as resources, schedules, goals, and objectives. The main purpose of this phase is to initiate the project, define the integrated system to be developed, create top-level design documents, and create a formal project plan or road map.

In creating your road map, you provide a detailed description of the integrated system to be developed. This plan serves the following purposes:

- Designing what your future system looks like
- Showing you the resource allocation needed to implement the design

If analysis is finding out where you are, planning tells you where you want to go and how to get there. When necessary, you can obtain help from SeeBeyond Consulting Services and other SeeBeyond representatives. Thorough, comprehensive planning helps to ensure a successful deployment project.

The major steps in deployment planning are:

- Determining overall objectives
- Initiation steps
- Creating the planning documents

3.3.1 Determining Overall Objectives

This step of the deployment planning phase includes the following operations:

- 1 Achieve a consensus on the project's overall functionality and scope by taking the following steps:
 - ♦ Set up technical and functional teams or roles to handle individual phases and aspects of the deployment.

Note: For a small deployment, one person could handle the tasks of a team.

- ♦ Ensure that the functionality is clearly stated and agreed upon.
 - ♦ Document the functionality and scope of the project based on analysis information, as well as match this information against the scope of the project as stated in the "Approved Proposal."
 - ♦ If necessary, resolve any differences between the "Approved Proposal" scope and your prepared analysis and requirements information (see "[Requirements Analysis](#)" on page 22).
- 2 Create a general model of what the system will do. This model:
 - ♦ Serves as the foundation architectural plan for all ICAN Suite design.
 - ♦ Consists of diagrams and supporting documentation that represents the design strategy for any required ICAN Suite interfaces.

- 3 Set up a design and development team and provide this team with an understanding of the application domain. Also provide the team with the top-level design documents.
- 4 Formulate a basis of validation of the final product during acceptance testing. This validation process includes the testing required to validate the functionality of the system and that it works as stated in the “Approved Proposal.”

3.3.2 Initiation Steps

The deployment project begins in earnest with the following tasks:

Hold a Project Kick-off Meeting

This meeting brings together all the members of the deployment project team. The analysis tasks and responsibilities assigned to each resource are also identified. The purpose of this task is to outline the reporting structure for the project and identify whom the Project Manager communicates with to ensure that other tasks in the project are being completed as planned. In addition, documentation standards are established.

Ensure That Hardware and Software Are Ready

The purpose of this task is to ensure that your hardware and software are in place and ready for the ICAN Suite installation. This process includes ensuring that the ICAN Suite software is fully supported on your hardware platform and operating system, and that the software has been shipped.

Complete Installation Test, Installation, and Checklist

The ICAN Suite installation task is completed during this phase to ensure that there are no issues with your technical environment. You can use a deployment checklist to detail the exact hardware and operating systems where the installation will be performed.

This task includes the following steps:

- The total ICAN Suite environment must be installed and tested. The deployment checklist is updated to identify what items were completed and to document any outstanding issues that prevented items from being completed.
- The hardware, software, and network requirements (current and planned) are identified and verified.
- The end-to-end communications with your other systems are also tested to ensure that communications are set up correctly and systems are exchanging messages correctly according to the communication protocol being invoked.
- Any communications with other businesses or trading partners are tested in the same way.

For detailed installation instructions, see the *SeeBeyond ICAN Suite Installation Guide*.

Establish the Change Management Process

Change management is a critical factor during all phases of the project. Change management identifies and tracks all changes for a project that depart from the original deployment plan. All changes must be identified and tracked because many small changes can impact a deployment project in the same way as a more easily identifiable large-scale change. Tracking all changes allows the Project Manager to plan and control a project and keep track of all changes in the project’s scope.

3.3.3 Creating the Planning Documents

In this step of the deployment planning phase, the following documents are created:

- Deployment project plan
- Functional requirements specification
- Technical requirements specification
- Initial test plan

Deployment Project Plan

The deployment project plan establishes a baseline reference plan. It is your road map for the deployment project. The roles and responsibilities of each organization, the schedule of tasks, and any estimates must be defined in this plan.

Be as detailed as possible. Any project risks must be assessed and documented. Your resources are budgeted using this plan (or validated if you have already created a budget).

The deployment project plan must be reviewed and agreed on by all of the organizations involved in the project. The plan must be communicated to all affected organizations.

Table 2 describes the contents of this document.

Table 2 Deployment Project Plan

Contents	Description
Scope of work	This item must be based on the purchase contract, “Approved Proposal,” or any equivalent document.
Project organization	Include the deployment project team (or the person responsible for a small deployment), development organization, review organization, and any external organizations involved in the project. The roles and responsibilities must be clearly defined.
Delivery schedule	Include the schedules for all specified deliverables, including the final delivery date after all validation and verification tasks are complete.
Estimates	This section includes a work breakdown structure (WBS).

Table 2 Deployment Project Plan (Continued)

Contents	Description
Overall schedule	This section contains a schedule for all deployment project tasks including resource assignments; key phase completion milestones must be indicated. This schedule will be further elaborated by developing various phase work plans (the use of Microsoft Project is recommended).
Resource requirements	Include the manpower, hardware, and software resources required for the pre-transition testing phase.
Issues and risks	Potential project issues and risks must be identified, and contingency plans must be drawn up for any risks.
Organizational interfaces	The dependencies on other projects and information needed from other organizations must be clearly identified, documented, and conveyed to any affected parties.

Functional Requirements Specification

In creating the functional requirements specification, you identify and analyze your specific system requirements. The behavior of the various application components (messages, process, and associated data) are carefully analyzed and documented.

The functional requirements specification and the technical requirements specification help form the basis for system design and final project acceptance.

The typical subtasks in creating this document are:

- Studying and identifying system requirements to derive the business process functionality required and identify the system architecture needed to meet functional requirements.
- Creating an architecture model to show the proposed integrated system.
- Creating interface models to define interface requirements.

Table 3 describes the contents of this document.

Table 3 Functional Requirements Specification

Contents	Description
Statement of requirements	Define the objectives that you want the project to meet.
Proposed architecture	Show a summary design model of the sending and receiving systems, eWay Intelligent Adapters to be used, and interfaces that take place.
Proposed directory structure and messages that trigger ICAN Suite processing	Provide a map of the external sending/receiving systems, directory structures, and the business/processing messages, including the ICAN Suite processing that will be initiated by the messages.
Exception processing	Define requirements for processing errors and exceptions.
Constraints	Define data volumes, performance, and any backup/archive requirements.

Table 3 Functional Requirements Specification (Continued)

Contents	Description
Interface diagrams	Produce a diagram for each proposed interface, showing the sending/receiving system, message processing, and any interdependencies.
Hardware/software diagrams	Show the hardware/software environment and high-level related schematics for development, testing, and production systems.

The general design model provided by this document forms the starting basis of the next deployment step, the system design and development phase. See [Chapter 4](#) for details on how to use this model as the foundation for your complete ICAN Suite architecture.

Technical Requirements Specification

In creating the technical requirements specification, you identify and analyze your specific technical requirements. The behavior of the various application components (messages, process, and associated data) are carefully analyzed and documented.

The functional requirements specification and the technical requirements specification help form the basis for system design and final project acceptance.

Table 4 describes the contents of this document.

Table 4 Technical Requirements Specification

Contents	Description
Hardware/software model	Defines the environment in which the ICAN Suite will process.
Security and system availability	Requirements for security, system availability, and the technology being used to meet these requirements.
Additional requirements	Any additional related requirements.

Initial Test Plan

A high-level test plan must be produced, highlighting the testing tasks to be performed during each phase. This document specifies the test approach, the types of tests to be carried out, and the organization responsible for carrying out the tests for each test phase.

Note: *The test plan is enhanced during the system design and development phase (see [Chapter 4](#)). The actual testing is carried out during the pre-transition testing phase (see [Chapter 5](#)).*

The initial test plan can be a single document, or it can consist of a separate document per project for all of the test phases, or one document per phase, depending on the size and complexity of the deployment project.

Table 5 describes the contents of this document.

Table 5 Initial Test Plan

Contents	Description
Test plan	Contains a general description of the testing phase. This plan is preferably produced by an independent test team or role based on your requirements for the testing of applications and their process for promoting applications to production.
Test phases	Includes unit testing, integration testing, and acceptance testing.
Test approach	Details whether there will be manual or automated testing and the validation process for each test performed.
Organization	Includes the testing team or role (functional and technical).
Schedule	Defines the system availability for test data and system resources needed for the different test phases.
Resource requirements	Defines system, individual, and team resources needed for the test phases.

Going Forward

When the members of your management and deployment project team (or other responsible persons) agree that these objectives have been met, you have finished the deployment planning phase. You have created a complete deployment road map (the deployment project plan), as well as some general designs for your completed system.

The remaining chapters in this guide treat the deployment project under the following topics:

- **System Design and Development:** For a discussion of the next phase of the deployment project, including system design architecture and development, see [Chapter 4](#). In this phase, you broaden and fill in the details of the general designs that you created during the planning phase.
- **Testing, Transition to Production, and Maintenance:** For a discussion of the pre-transition testing, transition to production, and post-transition maintenance phases, see [Chapter 5](#).
- **Hardware Needs:** An important part of planning for your system and deployment is how to determine your hardware requirements. If you need additional information on planning for and determining your system's hardware requirements, see [Chapter 6](#).

System Design and Development

This chapter describes the third phase of deploying the ICAN Suite: system design and development.

What's in This Chapter

- [“Introduction” on page 32](#)
- [“Service-Oriented Architecture” on page 34](#)
- [“System Design” on page 38](#)
- [“System Development” on page 39](#)
- [“Optimizing Your System” on page 46](#)

4.1 Introduction

When the requirements analysis and deployment planning phases have been completed, your next major step is the system design and development phase. In many ways, this is the heart of an ICAN Suite deployment. During this phase, you flesh out the essential system architecture that implements your business plans and processes.

This phase requires a series of successive refinements applied to your initial summary plan (the functional requirements specification). Your design must start with the broadest view of the system and then proceed to the details. This top-down approach results in the most effective application of the technology to the integration of your existing systems and applications.

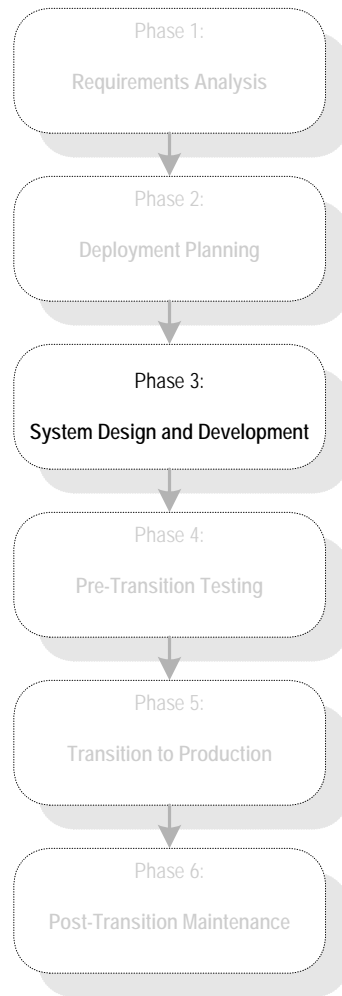
System design and development include the following steps:

- Planning general hardware configuration
- System design methodology
- Software development
- System optimization

These steps are not necessarily an exact sequence. You may need to “back-track” to earlier steps and look forward to later steps, to implement the correct design refinements that your system requires.

Figure 6 shows where the system design and development phase fits into the overall ICAN Suite deployment.

Figure 6 System Design and Development Phase



Because you can distribute an ICAN Suite system over as many hosts as you need to provide sufficient computing power, this chapter guides the decisions that you must make to deploy an effective ICAN Suite environment. The chapter also presents a methodology for designing an ICAN Suite environment.

4.2 Service-Oriented Architecture

The ICAN Suite enables you to develop composite applications on top of a *service-oriented architecture*.

4.2.1 Overview

In a service-oriented architecture, a group of software components offers functionality to other components. A service consumer makes a request, and a service provider sends a response. Some services are high level (for example, verify the insurance eligibility of a patient). Other services are more technical (for example, transform a message from an external format to an internal format).

The service provider publishes an interface that service consumers interact with. The implementation details are hidden from the consumers.

In the ICAN Suite, services can be of varying types, including:

- Java Collaborations
- Business Processes
- Page Flows
- eTL Collaborations

These services are created independently in Enterprise Designer and stored in the Repository. You then use a Connectivity Map to link the services and other component types (such as topics and external applications) together.

Figure 7 shows a simple Connectivity Map that contains two services, both of which are Java Collaborations.

Figure 7 Connectivity Map with Two Services



One of the key benefits of a service-oriented architecture is reusability. Services can be combined in different ways. For example, Service1 in the preceding Connectivity Map may also appear in a different Project.

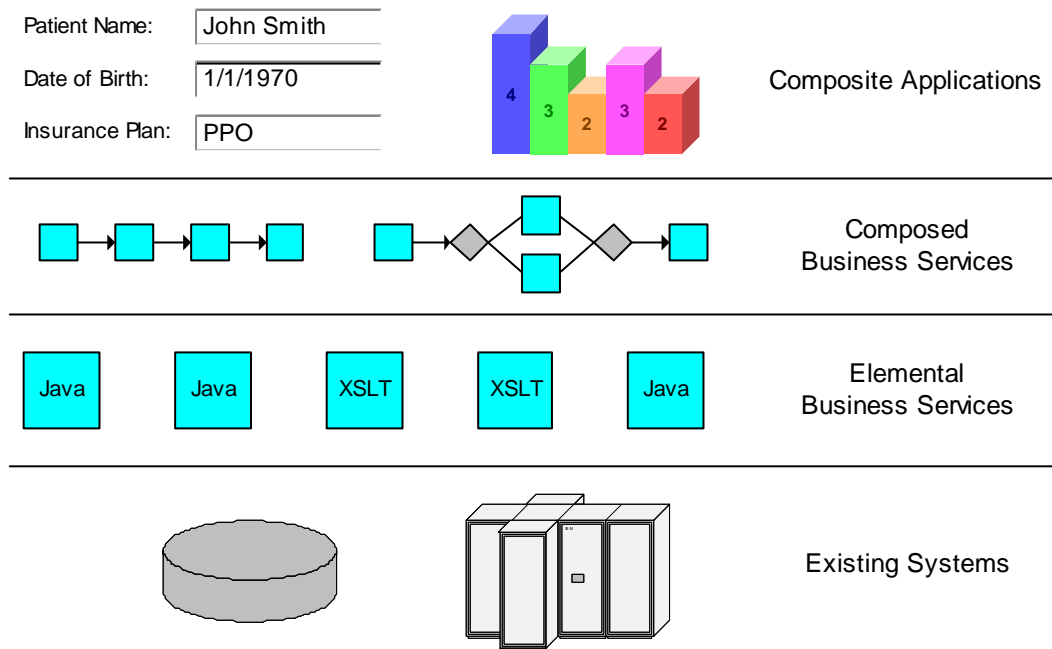
Another benefit is location transparency. The service consumer does not need to know where the service provider is located on the network, and vice versa.

A third benefit is that services in a service-oriented architecture are loosely coupled. This concept means that changing a service does not necessarily require changes to the components with which it interacts.

4.2.2 Architecture Layers

Figure 8 shows the four layers of the ICAN Suite’s service-oriented architecture:

Figure 8 Architecture Layers



The following subsections describe each layer, starting with the lowest layer and moving upward.

Existing Systems

The Existing Systems layer consists of back-end programs, such as databases, packaged applications, external trading partners, and legacy systems.

The back-end programs may use technologies that are incompatible with each other. For example, information about a customer may be located in an Oracle database (used by one company division) and in an IBM DB2 database (used by another company division).

Elemental Business Services

The Elemental Business Services layer contains services that perform discrete, fine-grained tasks, such as:

- Verify customer credit
- Determine product availability
- Calculate shipping charges

eGate Integrator enables you to define these services, which can be based on Java or XSLT. Object Type Definitions (OTDs) define the various formats of messages that the services process.

eTL Integrator enables you to extract, transform, and load bulk data between files and databases.

eView Studio provides methods that you can use in Java Collaborations to specify how data is processed into the master index database.

eWay Intelligent Adapters allow the services to interact with the different technologies in the Existing Systems layer. Thus, the adapters shield the incompatibilities between back-end programs from this layer.

Composed Business Services

The Composed Business Services layer combines services in the lower layer into larger-grained services that represent a specific business process. Examples of services in this layer are:

- Process customer order
- Execute revenue process

In many scenarios, conditional branching is required. An example of conditional branching is: if condition 1 is true, invoke service 1; otherwise, invoke service 2. The second Composed Business Service in Figure 8 includes a conditional branch.

You can incorporate human tasks into the flow of a business process. The flow waits for a person to complete the task before proceeding further.

Some business processes are long running, which means that under normal circumstances they can take days, weeks, or months to complete.

eInsight Business Process Manager contains a wide variety of design elements (including elements that support conditional branching) for creating business processes in this layer.

Note: You can also use eInsight to create an Elemental Business Service.

eView Studio enables eInsight to access the master index database.

eXchange Integrator includes the concept of a B2B protocol, which represents a business process in trading partner scenarios.

Composite Applications

The Composite Applications layer contains applications that are based on existing applications and data.

The information that appears in the user interface may be combined from multiple back-end systems. Without the composite application, the user would need to obtain the desired information by accessing each system independently.

eVision Studio enables you to create user interfaces for composite applications.

The following ICAN Suite products are available for creating specialized user interfaces:

- eBAM Studio enables you to create dashboards that illustrate Key Performance Indicators (KPIs).
- ePortal Composer enables you to create Web portals that present a single, unified view of applications and data.

eIndex Global Identifier is a composite application included with the ICAN Suite. It enables you to create a single view of person information across an enterprise.

4.2.3 Invoking Services

By default, the ICAN Suite uses Web services technology to invoke services. Table 6 describes the main Web service standards:

Table 6 Web Service Standards

Standard	Description
Extensible Markup Language (XML)	A form of Standard Generalized Markup Language (SGML) that is widely used for data exchange and configuration files.
Simple Object Access Protocol (SOAP)	An XML-based specification for exchanging messages in a distributed environment, typically over HTTP.
Web Services Description Language (WSDL)	An XML-based language for describing the functionality that a Web service offers.
Universal Description, Discovery, and Integration (UDDI)	A specification that enables Web services to be published in a registry. Service consumers can search the registry for service providers.

The ICAN Suite provides alternate methods for service invocation, such as Java and COM.

4.3 System Design

The three basic steps in designing an ICAN Suite deployment are:

- Identify all of the external systems to be connected
- Define the configuration of ICAN Suite components
- Define the configuration of hardware and network connections

4.3.1 Identifying External Systems

The first step is to identify all of the external systems to be connected to each other through the system. The resulting set of interconnected systems is called the *communication topology*. The communication topology exists without regard for the hardware hosts where components execute and without regard for the format of data exchanged between systems.

4.3.2 Configuring ICAN Suite Components

The second step is to define a configuration of ICAN Suite components (for example, eWay Intelligent Adapters, Collaborations, and Message Servers) to run on the respective hosts in the hardware topology. The component arrangement is called the *component topology*.

Choose the simplest Object Type Definitions and Collaborations. Defining the most efficient component topology depends on the relationship of data formats. Therefore, defining object types is an integral part of designing the component topology.

4.3.3 Hardware and Network Connections

The third step is to define a configuration of hardware and network connections that enable the external systems to communicate as required by the communications topology. This hardware configuration is called the *hardware topology*.

The ICAN Suite is designed to run as a distributed system with central management. Only network performance and the demands on each host are relevant considerations in defining hardware topology. Because of the distributed architecture of the total system, the hardware topology is not rigidly defined. It can be adjusted as needed when system demands change. For example, increased demands on the ICAN Suite environment can be met by distributing processing across more CPUs.

For more information on how to determine and meet your hardware requirements, see [Chapter 6, “Determining System Requirements”](#).

4.4 System Development

This section covers a variety of system development topics, such as user management, naming conventions, and project teams with multiple developers.

4.4.1 Creating Users

The ICAN Suite has one predefined user: **Administrator**.

The **Administrator** user has unlimited power to create other users. Each user has one or more *roles*, which specify the tasks that the user can perform. The ICAN Suite includes the following predefined roles:

- all
- administration
- management

Every user has the **all** role. The **administration** and **management** roles provide higher levels of privileges.

When creating a user, the **Administrator** user should not give the user a more powerful role than necessary.

Once the developer users have been created, they can use Enterprise Designer to create the Project and Environment components.

If your organization stores information about the project team members in an LDAP server, you can integrate with the LDAP server. In this release, the following LDAP servers are supported: Microsoft's Active Directory, Sun Microsystems' Sun Java System Directory Server, and OpenLDAP.

***Note:** The type of user management described in this section is called Repository User Management. The term Environment User Management refers to the management of users who access deployed Projects. For detailed information about both types of user management, see the "ICAN Security Features" chapter in the eGate Integrator System Administration Guide.*

4.4.2 Multiple Repositories

If the production system will be run on a separate network from the development system, you may want to create one Repository for the development phase and another Repository for the production phase. This approach makes it easier to use the Version Control system and to monitor the production system.

If the development and testing will be run on the same network, you can use the same Repository for development and testing.

4.4.3 Naming Conventions

SeeBeyond recommends that you create a set of naming conventions early in the system design and development phase. Consider using the prefixes and suffixes described in this section. Additional standards may include:

- Start the main portion of a Service name with a verb
- If applicable, include the data format or external system in an Object Type Definition name.

Prefixes

Add a prefix to each component name to identify the component type. For example, you could use the prefix **prj** as an identifier for Projects.

Table 7 contains suggested prefixes for Project components.

Table 7 Suggested Prefixes for Project Components

Component	Prefix	Examples
Project	prj	prjFileTransfer prjWebOrder
Business Process	bp	bpProvideQuote
Collaboration Definition (Java)	jcd	jcdOrderIn jcdCheckInventory jcdOrderOut
Collaboration Definition (XSLT)	xcd	xcdOrderIn xcdCheckInventory xcdOrderOut
Connectivity Map	cm	cmProvideQuote cmWebOrder
Deployment Profile	dp	dpWebOrderDev dpWebOrderTest dpWebOrderProd
External Application	ea	eaFileIn eaFileOut
Object Type Definition	otd	otdEmployeeInfo otdOraOrders
Page Flow	pf	pfVacationRequest
Page Layout	pg	pgCredit
Page Link	pl	plMortgage
Queue	que	queOraOrders
Service	svc	svcTransfer svcReadFromFile svcWriteToFile
Topic	tpc	tpcReqRplInput

If you create subprojects as an organizational aid (for example, as the container for a large number of Collaboration Definitions or Object Type Definitions), then you can omit the **prj** prefix. See Figure 9.

Figure 9 Naming Subprojects

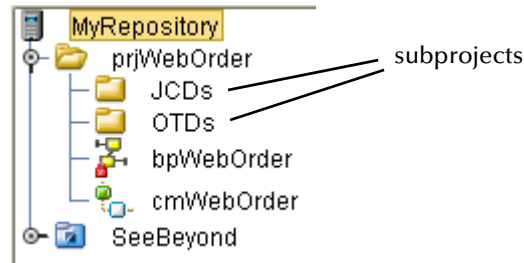


Table 8 contains suggested prefixes for Environment components.

Table 8 Suggested Prefixes for Environment Components

Component	Prefix	Examples
Environment	env	envWebOrderDev envWebOrderTest envWebOrderProd
Logical Host	lh	lhWebOrderDev lhWebOrderTest lhWebOrderProd
Integration Server	is	isWebOrderDev isWebOrderTest isWebOrderProd
External System	es	esFileIn esFileOut

Suffixes

You may want to add a suffix to components that have development, test, and production versions. In Table 7, the examples for the Deployment Profile use this convention. In Table 8, the examples for the Environment, Logical Host, and Integration Server use this convention.

You can also use suffixes to indicate whether External Applications and External Systems are inbound or outbound.

4.4.4 Project Teams with Multiple Developers

This section contains guidelines for project teams with multiple developers.

Subprojects and Multiple Projects

If you have a large development team, you may find it useful to divide the work using one of the following approaches:

- Create multiple subprojects within a top-level Project. Choose this approach when the subprojects have common components.
- Create multiple Projects.

When using the first approach, the developers must consider the order in which components are created. For example, assume that a Collaboration Definition called **jcdCheckInventory** is used by two subprojects. It would make sense to work on this Collaboration Definition before working on Collaboration Definitions that are not used by multiple subprojects.

Version Control and Access Control Lists

The Version Control system and Access Control Lists (ACLs) allow a project team to limit access to components as needed.

With the Version Control system, a developer can work on the new version of a component while other team members use the most recent stable version. For example, assume that version 1.4 of an Object Type Definition called **otdEmployeeInfo** is considered to be stable. While the developer works on version 1.5, everyone else can use version 1.4.

Important: *More than one person concurrently using the same user name will circumvent the Version Control system, and one person's work can be overwritten by another. Ensure that all personnel using Enterprise Designer use unique names.*

ACLs become more important as the size of a Project grows, or when you divide the work as described in “**Subprojects and Multiple Projects**”. They allow you to control which users have read only or read/write access to a component, and thus reduce the possibility of deliberate or accidental changes.

Ensure that the development team fully understands how the Version Control system and ACLs work. The *eGate Integrator User's Guide* describes how to use the Version Control system. The *eGate Integrator System Administration Guide* describes how to create and modify ACLs. The ICAN Suite readme file may contain additional information.

Activating Multiple Deployment Profiles

As explained in the *eGate Integrator User's Guide*, a Project can have more than one Deployment Profile. However, only one of the Project's Deployment Profiles can be active in a single Environment.

If your development team needs to be able to activate multiple Deployment Profiles at the same time, then either create multiple development Environments or create a Connectivity Map and a Deployment Profile in a subproject.

Logical Hosts

Ideally, each developer installs and uses a Logical Host on the same computer as Enterprise Designer. Thus, developers can run the Logical Host without interfering with the work of other developers.

If a development computer does not have enough resources to run a Logical Host, you can try other approaches.

Note: Typically, the amount of system resources used by a developer's Logical Host is less than the amount of system resources used by a Logical Host in the test and production environments. **Chapter 6, "Determining System Requirements"** lists the minimum system requirements for the Logical Host on various operating systems.

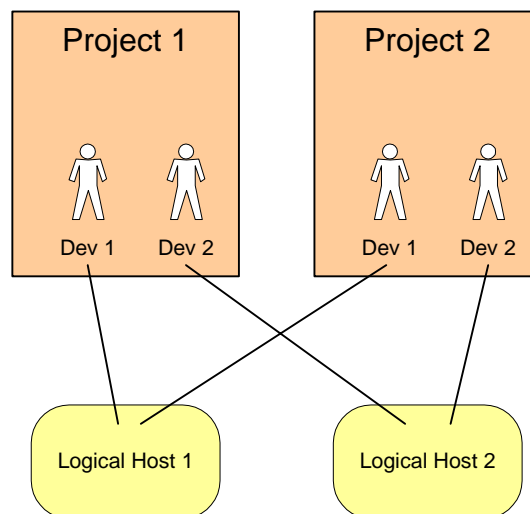
One approach is to share Logical Hosts. For example, assume that there are two Projects, each of which has two developers. Rather than creating four Logical Hosts (one for each developer), the teams could create two Logical Hosts:

- Logical Host 1 is shared by developer 1 from the first project and developer 1 from the second project.
- Logical Host 2 is shared by developer 2 from the first project and developer 2 from the second project.

You should use this approach only when the two projects are sufficiently independent from each other.

Figure 10 illustrates sharing Logical Hosts between Projects.

Figure 10 Sharing Logical Hosts Between Projects



4.4.5 Error Handling

In the requirements analysis and deployment planning phases, you defined the requirements for processing errors and exceptions.

In the system design and development phase, you use various ICAN Suite features to implement these requirements. For example:

- When creating a business process in eInsight, you can configure the process to catch all exceptions or certain exceptions that you specify. In addition, eInsight provides a Compensation Handler element, which allows you to add sophisticated rollback logic to Straight Through Processing (STP), fast processes, and long-running transactions.
- eVision allows you to return preconfigured Web pages if a requested page cannot be found or if there is a serious problem with the business process. These preconfigured Web pages correspond to the HTTP response codes 404 and 500, respectively.

4.4.6 Variables and Constants

Variables and constants enable you to avoid hard-coding environment-specific information such as file names and database names.

A variable functions as a placeholder. The value is determined when you create a Deployment Profile. You can define variables at the Project level.

A constant is a name/value pair that is visible across a Project. You can define constants for a Project or an Environment.

Assume that you create a Connectivity Map that includes an inbound File eWay. The eWay has a property called **Directory**, which specifies the folder that the eWay polls for input files. Instead of entering a value such as **C:/input**, you can select a variable that you have previously defined (for example, **VARIABLE_FILE_INPUT**). When you create the Deployment Profile, you assign a value to the variable.

For more information, see the *eGate Integrator User's Guide*.

4.4.7 Multiple Environments

SeeBeyond recommends that you create separate Environments for the development, test, and production phases.

Each version of the Environment should have a corresponding Deployment Profile:

- The development Deployment Profile maps Project components to the development Environment.
- The test Deployment Profile maps Project components to the test Environment.
- The production Deployment Profile maps Project components to the production Environment.

4.4.8 Creating Multiple Run-Time Components

Under normal circumstances, you should not need to create multiple run-time components. In other words, you can run a single Logical Host that contains a single SeeBeyond Integration Server and/or a single SeeBeyond JMS IQ Manager.

The Integration Server is designed to scale up to meet higher demands. Note that you can modify Integration Server properties associated with scalability (such as the **Maximum Size of Message-Driven Bean Pool** property) from Enterprise Designer.

However, there are situations when you may want to create multiple run-time components.

Assume that you are running a Logical Host on a computer that has multiple CPUs. If you discover that some of the CPUs are being underutilized, check to see whether any of the following steps improves performance:

- Create an additional Integration Server within the Logical Host
- Create an additional JMS IQ Manager within the Logical Host
- Create an additional Logical Host

Multiple Logical Hosts can be useful in isolating deployed Projects from each other. For example, if you are running two Projects on the same Logical Host and Project 1 requires the Logical Host to be restarted, then Project 2 is also restarted. Using multiple Logical Hosts would prevent Project 2 from being restarted.

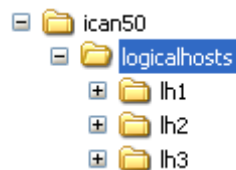
In general, be sure to test and monitor your deployed Projects so you can determine how effectively the run-time components are handling the load.

4.4.9 Directory Structure for Multiple Logical Hosts

If you install multiple Logical Hosts on the same computer, SeeBeyond recommends that you place them in a common Logical Host directory.

Figure 11 shows an example of this structure on a Windows computer. First, you create the **C:\ican50\logicalhosts** directory. You then install the first Logical Host into this directory. To distinguish the first Logical Host from the other Logical Hosts that will be installed, you rename the **C:\ican50\logicalhosts\logicalhost** subdirectory to **C:\ican50\logicalhosts\lh1**. For the second and third Logical Hosts, you rename the subdirectories to **C:\ican50\logicalhosts\lh2** and **C:\ican50\logicalhosts\lh3**, respectively.

Figure 11 Multiple Logical Hosts on a Single Computer (Windows Example)



Here is an example of this structure on a UNIX computer:

```
/home/ican50/logicalhosts/lh1  
/home/ican50/logicalhosts/lh2  
/home/ican50/logicalhosts/lh3
```

If you plan to run more than one Logical Host at the same time, be sure to configure the base port numbers so that they do not conflict with each other. For more information, see the “Logical Hosts” chapter in the *eGate Integrator System Administration Guide*.

4.5 Optimizing Your System

Optimizing your system depends on such factors as:

- The size of your Project
- The complexity of the Object Type Definitions
- Which ICAN Suite products you are using

4.5.1 Project Size

To achieve the best performance for a small Project, SeeBeyond recommends the following JMS client and Integration Server settings. These recommendations are independent of operating system, CPU, and memory. You can set all of the values from Enterprise Designer.

Table 9 JMS Client Recommended Setting (Small Project)

Property	Value
Server session pool size	20

For more information about the **Server session pool size** property, see the “Configuring JMS Clients” chapter in the *eGate Integrator JMS Reference Guide*.

Table 10 Integration Server Recommended Settings (Small Project)

Property	Value
Minimum Size for Thread Pool	30
Maximum Size for Thread Pool	30
Minimum Size of Message-Driven Bean Pool	20
Maximum Size of Message-Driven Bean Pool	50
Minimum Size of Stateless Session Bean Pool	0
Maximum Size of Stateless Session Bean Pool	50

For more information about these properties, see the “Environments” chapter in the *eGate Integrator User’s Guide*.

Testing, Transition to Production, and Maintenance

This chapter describes the final three phases of deploying the ICAN Suite: pre-transition testing, transition to production, and post-transition maintenance.

What's in This Chapter

- [“Introduction” on page 47](#)
- [“Pre-Transition Testing” on page 49](#)
- [“Transition to Production” on page 54](#)
- [“Post-Transition Maintenance” on page 54](#)
- [“Summary” on page 56](#)

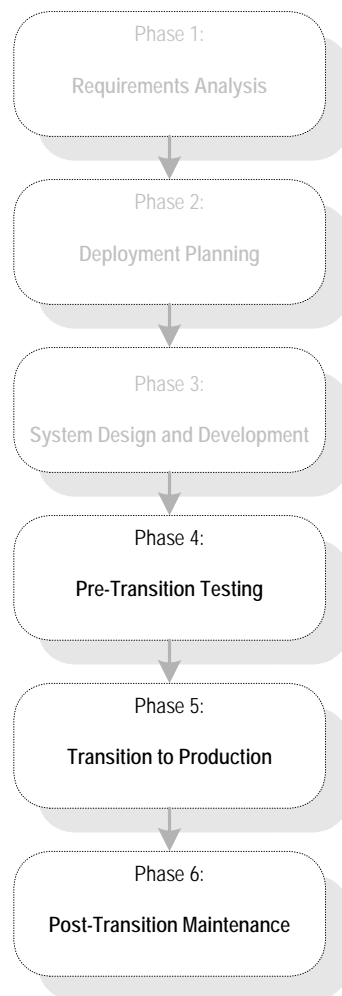
5.1 Introduction

When the system design and development phase has been completed, there are three remaining phases:

- **Pre-transition testing:** The system must be fully tested before it is transitioned to a production environment. This phase includes unit testing, integration testing, and acceptance testing.
- **Transition to production:** After the system is fully tested, it must be transitioned, or migrated, to the production environment. This chapter covers the procedures and considerations for performing the transition to production.
- **Post-transition maintenance:** Once the system has been migrated to the production environment, the system must be monitored for correct performance, possible errors, and the need for changes. System monitoring is a critical step in the long-term success of your deployment project. Routine checks and fine-tuning help to establish long-term performance benchmarks and aid in identifying undesirable changes.

Figure 12 shows where these phases fit into the overall ICAN Suite deployment.

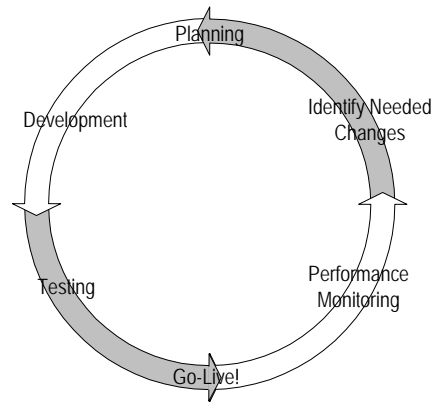
Figure 12 Testing, Transition to Production, and Maintenance Phases



An important part of the entire deployment project is change management. If changes are required, they must be processed through the same cycle of planning, development, testing, transition to production, and maintenance as the rest of the deployment.

Figure 13 illustrates this cycle of change management.

Figure 13 Change Management Cycle



5.2 Pre-Transition Testing

An essential part of the implementation of any complicated system is thorough testing. Table 11 describes the types of testing that must be performed:

Table 11 Testing Types

Type	Description
Unit testing	Testing of individual components in isolation.
Integration testing	Testing of groups of components together, up to and including the entire system.
Acceptance testing	Testing of a completed system (or portion thereof) to ensure that it meets the requirements.

For the most part, unit and integration testing are done in the development phase, while acceptance testing is done as a final check before putting the system into production.

5.2.1 Testing Methodology

While how a system is tested varies, depending on the particulars of the specific system, certain methodologies apply to all system testing.

In general, you test the individual parts of the system before testing the entire system. Similarly, you test individual components in isolation before testing them in a broader context.

5.2.2 Test Plan

Planning for system testing begins with a careful examination of the requirements of the system. A test plan is created in the deployment planning phase. This test plan specifies how the system is tested and what requirements the system must meet before it is put into production (see Chapter 3).

The test plan is further refined in the design phase.

The functional and technical specifications outline the exact procedure used to conduct the tests, both at a component level and at an integrated system level. These specifications include:

- Type of data to use
- Expected output
- Who is responsible for the test

Type of Data To Use

The test plan specifies the type of data to use when testing the system. At both the component level and the integration level, be sure to work with data that is typical of the data that the system is designed to process. If possible, use real data from your pre-existing systems. Vary the data enough so that all possible types of processing implemented by the system are tested.

In addition to real-life typical data, use data designed to test the system's error handling. This data may have to be specially constructed.

Testing the Output

The test plan includes specifications for:

- Proper error handling
- Transaction processing speed
- Correct routing of information
- Correct transformation of data
- Any other special requirements

Responsibility for Testing

In general, the responsibility for testing an individual component belongs to the developer of the component. The responsibility for the testing of the entire system may belong to the project manager or the technical lead for the project.

Acceptance testing is done by or in conjunction with people for whom the system is being created. Often this is the person or persons who are using the system when it is put in production.

5.2.3 Unit Testing

Unit testing checks the individual parts of a larger system for correct functioning prior to integration testing.

Each component used in the system must be unit-tested and its functionality verified before the component can be used in the integrated system.

Unit testing is done as part of the development phase by the developer responsible for creating the component. If the functional or technical specifications include a procedure for testing a particular component, this procedure must be followed. If the functional and technical specifications do not specify test procedures for a component, at a minimum the test must verify that the component performs as outlined in the specifications, testing individual parts of a system.

The tools used to test each part are different, depending on what the part is designed to do in the system. In addition, though all are designed to verify correct functioning, the methods employed vary, depending on the component being tested.

Take advantage of the validation tools in various ICAN Suite products. For example, Enterprise Designer includes a Java Debugger that enables you to debug Java-based Collaboration Definitions.

5.2.4 Integration Testing

Integration testing verifies how well the new components work together and with the existing ICAN Suite's infrastructure.

If the system is large and complex, you can break the integration testing into pieces designed to test a self-contained portion of the system.

Partial Integration Testing

Partial integration testing verifies correct data movement from one external system to another (that is, a complete data path).

For example, assume that you have a system that brings in data from a single external system and then sends the data to several other external systems. A partial test of this system would be to test whether the data can be sent to one of the external systems.

Complete System Testing

Complete system testing tests the entire system, including the interaction with all of the external systems. If the system is large and complex, this type of test requires a large amount of coordination. Set up this test to duplicate the actual production system. In many cases, it is used as a dry run prior to doing the actual acceptance test.

Performance Testing

Integration testing tests whether the system works; performance testing tests whether the system works fast enough. This type of testing must be done once a component or system is functioning correctly and transforming the data properly. It is important that you do this type of testing in the context of integration testing, because many factors in combination affect performance. Speeding up one component may not speed up the performance of the entire system if the bottleneck lies elsewhere.

The exact requirement or goal in terms of the system's performance must be specified in the test plan. Whether you meet the goal determines whether you pass the test.

An additional goal in performance testing is to find the bottlenecks in the system. Uncovering these bottlenecks allows additional system resources to be allocated intelligently in order to improve processing.

Speed Testing

This operation tests whether the system processes data fast enough. Make sure that the logging is set to the levels that will be used in production before doing a speed test. Higher-than-normal levels of logging can seriously degrade system performance and slow processing speed.

Stress Testing

This operation tests whether the system can handle the expected load. This type of testing attempts to overload the system with data to see whether or how it could fail. Effective stress tests include large volumes of messages, large-sized messages, and multiple concurrent connections. Many times, network bottlenecks can be uncovered this way. The test plan describes the methodology to employ for this type of test.

5.2.5 Acceptance Testing

Acceptance testing is done before moving the system into production. It is used as a final check to prove to its end users that it performs according to plan.

Acceptance testing can be done for part of a system or for a complete system. If the entire system is not going into production at the same time, you can test the portion of the system that is going into production.

The test plan specifies all conditions that the system has to meet before being put into production. In addition, the test plan specifies the people who must approve the system and must be involved in the test.

5.2.6 Troubleshooting

The following tools are available for finding and correcting errors:

- ICAN Monitor
- Log files created by individual components
- Alert Agent
- SNMP Agent

ICAN Monitor

The ICAN Monitor gives you control over the components in a Project. The ICAN Monitor alerts you to the status of components (for example, whether they are running) and allows you to send commands to them (such as start or shut down).

For more information on using the ICAN Monitor, see the following documents:

- *eGate Integrator System Administration Guide*
- *eGate Integrator JMS Reference Guide*
- *eInsight Business Process Manager User's Guide*

Log Files

To gain additional information, use the log files created by any component that fails.

When a component or system is not working, errors are written to the log file to help you diagnose the problem. In a normally functioning system, only the most serious errors (such as a shut-down component) are written to the log, because each entry written to the log uses system resources and slows processing. To learn more about a malfunctioning component, you can enable the system to log more information.

For more information on log files, see the "Monitoring Logs" chapter in the *eGate Integrator System Administration Guide*.

Alert Agent

The Alert Agent enables you to send a specified category of Alerts to one or more destinations as the Alerts occur. For detailed information, see the *Alert Agent User's Guide*.

SNMP Agent

The SNMP Agent enables you to forward alerts as SNMP version 2 traps to a third-party SNMP management system. For detailed information, see the *SNMP Agent User's Guide*.

5.3 Transition to Production

After verifying the performance and reliability of the system, the next step is to transition it to the live production environment.

This phase consists of the following steps:

- 1 Create a production version of the Deployment Profile
- 2 Activate the Deployment Profile
- 3 Start the Logical Host

The *eGate Integrator User's Guide* describes the first two steps. The *eGate Integrator System Administration Guide* describes the third step.

5.4 Post-Transition Maintenance

With your project up and running, it is important to monitor system activity, check for errors, and make any needed changes.

Figure 13 on page 49 shows the steps to take when changes are required. You must put changes through the same high scrutiny as the original system design.

5.4.1 Monitoring System Activity

Monitoring system activity is a critical step in the long-term success of the system. Routine checks help to establish long-term performance benchmarks. Such benchmarks are helpful in identifying undesirable changes. If you do not have a good feel for a healthy system's vital signs, you may have a more difficult time recognizing when your system needs attention.

ICAN Monitor

The ICAN Monitor provides visual cues to let you know when a component needs attention. For example, the Connectivity Map in the Project Explorer displays a flashing red square when a Service becomes inactive.

Note: *The timeout interval for Enterprise Manager is three hours.*

If you configure the Alert Agent or SNMP Agent, you can avoid having to run the ICAN Monitor continuously. The agent will notify you when the specified problem occurs.

Log Files

On a daily basis, review the run-time log files and examine any messages with a severity level of FATAL, ERROR, or WARN.

Note: *You can specify the minimum severity level that a logger outputs to the corresponding log file. Do not use the DEBUG level.*

Many of the log files (such as **repository.log**) are configured to behave as a recirculating stack. Each file has a maximum size, and the oldest file is reused when the maximum number of files is reached. The ICAN Suite includes configuration files that enable you to change the maximum file size and the maximum number of files.

The other log files (such as **connection.log**) do not have configuration files. On a regular basis, check the size of these log files. If a file has grown too large, you can rename the file, delete the file, or move the file to a separate disk or computer.

Repository Backup

On a regular basis, back up the Repository. The frequency will vary depending on the project. One scenario is to back up the Repository once a week at a time of low system activity.

5.4.2 Implementing Changes

After a period of time, you may need to make changes to your project. Changes are common as the needs of your end users evolve and as additional external systems are added.

Do not make changes to the system hastily. Handle changes using the same process that was originally used to deploy your project. Consider the change management process illustrated in **Figure 13 on page 49**. Applying this same process of planning, configuration, testing, migration, monitoring, and re-evaluation will help ensure a successful deployment.

5.5 Summary

The proper use of a lab environment provides an excellent opportunity to verify and refine the system configuration that was implemented earlier in the deployment process. Focusing on the deployment plan ensures the smoothest possible deployment.

By thoroughly unit testing and system testing the system in the lab, costly errors are avoided and the end users are more satisfied with the final results.

Going Forward

Once all the members of your management team, your deployment project team, and the end users agree that the system is up and running according to plan, you have successfully finished the deployment.

If you have any questions about further system operation and maintenance, see the appropriate documents listed in [“Related Documents” on page 12](#) or contact SeeBeyond.

Determining System Requirements

This chapter offers guidelines to help you determine the system requirements for the deployment of the ICAN Suite.

What's in This Chapter

- [“Introduction” on page 57](#)
- [“Initial Considerations” on page 58](#)
- [“Estimating Requirements” on page 58](#)

6.1 Introduction

This chapter explains how to assess your needs for the following types of hardware:

- CPUs
- Random access memory (RAM)
- Hard disk space

There are many factors to consider in order to determine the hardware requirements for your particular system. As such, this discussion is limited to issues as they relate directly to the ICAN Suite.

This chapter does not consider networking topology, and does not address such issues as shared applications, how resources are distributed throughout a network, and how many workstations are included in the network. In the case of databases, it is assumed that each database management system is installed on a separate host.

6.2 Initial Considerations

Depending on the number of external connections, the type of data being processed, and how the data is processed, the required resources can vary. Take the following points into account as you begin estimating your hardware requirements:

- Each ICAN Suite deployment is different. The configuration of each deployment is unique because there are varying numbers of components as well as variances in interconnectivity. Some components are bidirectional and complex, while others merely pass data through.
- In addition to differences in configuration, the computational requirements will differ both in scope and complexity. The high-performance architecture of the ICAN Suite is net-centric, which makes the ICAN Suite highly flexible. It is this flexibility that makes predicting the general requirements of hardware a complex task.
- An ICAN Suite solution is distributed using run-time components and is platform independent. System stability and redundancy are important considerations. Server requirements vary greatly, depending on the components resident on the server, the archiving requirements configured on the customer's system, and other factors.
- Instead of only referring to absolute minimum requirements, it is more meaningful to discuss the hardware needs of an installation in terms of recommendations. By using the methods described in this chapter, a system analyst can estimate the required resources for a given configuration, from a simple deployment to a complex deployment, and thereby define an initial recommendation. Once installed, the ICAN Suite can be fine-tuned, both in terms of hardware and software, to optimize performance.

6.3 Estimating Requirements

This section lists some of the factors that affect performance, and then provides guidelines for estimating requirements.

6.3.1 Factors

Because the ICAN Suite is a general-purpose toolkit that is flexible in its deployment and configuration, estimating the processor requirements is a challenge. There are infinite possibilities and numerous factors with complex interactions that affect the estimates. Some factors are more critical than others, depending on the circumstances.

Some of the factors that affect performance are:

- CPU type and architecture
- CPU speed

- Presence of a CPU cache and its size
- Number of CPUs
- Physical memory size
- Swap size
- Disk subsystem (that is, bandwidth, latency, block size, RPM, seek time, and the presence and size of the cache)
- Network bandwidth and load
- Number of external systems and their latencies in servicing messages and acknowledgements
- Complexity and amount of processing to be performed by each component
- Message volume, size, and distribution through the day
- Throughput and response-time requirements
- Complexity of messages, including the number of nodes and complexity of regular expressions
- Bundling of messages (that is, more than one logical record in one physical record)
- Number of transitions between components for a given message (for example, moving data from an eWay to a topic/queue to an eWay or Collaboration)
- Amount of the implementation that can utilize parallel processing
- Other loads on the Logical Hosts (for example, queue cleanup schedules, backups, and other processes)
- Dispersion of solution across multiple CPUs and systems
- Number and architecture of subcomponents participating in the Project

6.3.2 Guidelines

A pragmatic approach is to start with a good base configuration as a development system and use that configuration to predict the processor requirements for a production system for your unique implementation.

The recommended methodology is to implement a representative number of interfaces (that exercise a good sample of the various data transformations and communication requirements of the implementation) on this system, run representative data files through the system, and record the CPU load. From these measures, you can project what the final production load will be and, therefore, the CPU requirement. Of course, the architecture can be tuned to achieve more efficiency using this same technique.

One of the advantages of the ICAN Suite's distributed and scalable architecture is that hardware does not need to be replaced but can be included in a multi-system implementation. Therefore, as processing requirements grow, you can easily add new hardware.

General Considerations

When estimating resource requirements, certain general considerations exist.

Table 12 lists the disk space and RAM considerations for the Repository, Logical Host, and Enterprise Designer machines.

Table 12 General Considerations

Machine Role	Disk Space	RAM
Repository	The disk space usage increases as additional ICAN products are installed.	Memory usage increases with each additional client connection. Every Logical Host and Enterprise Designer connection uses RAM resources on the Repository machine.
Logical Host	<p>Disk resources are used only by the products deployed on a Logical Host.</p> <p>A Logical Host running only one component will use less disk space than a Logical Host that contains multiple configured products.</p>	<p>Memory usage increases when:</p> <ul style="list-style-type: none"> ▪ The number of Collaborations and other deployed modules increases. ▪ The size and complexity of messages increase.
Enterprise Designer	The disk space usage increases as additional ICAN products are installed.	Memory usage increases based on the number of projects or components being configured.

Minimum System Requirements

The following tables list the minimum system requirements for ICAN Suite systems. The RAM and disk space requirements do not take into consideration the RAM and disk space required by the operating system. Note that the requirements vary based on the operating system and the role of the host machine.

Note: Larger Projects may require up to 1 GB of RAM for Enterprise Designer. For optimum performance, SeeBeyond recommends that you do not run other applications at the same time as Enterprise Designer.

Table 13 Windows System Requirements

Component	CPU	RAM	Disk Space
Enterprise Designer	1.2 GHz Pentium class	512 MB*	250 MB
*The minimum is 512 MB but the recommended is 1 GB.			

Table 14 Windows Repository and Logical Host System Requirements

Component	CPU	RAM	Disk Space
Repository	1.2 GHz Pentium class	512 MB	1.2 GB
Logical Host	1.2 GHz Pentium class	512 MB	250 MB

Table 15 HP Tru64 System Requirements

Component	CPU	RAM	Disk Space
Repository	667 MHz	512 MB	1000 MB
Logical Host	667 MHz	512 MB	350 MB

Table 16 HP-UX System Requirements

Component	CPU	RAM	Disk Space
Repository	540 MHz	512 MB	1150 MB
Logical Host	540 MHz	1.5 GB	500 MB

Table 17 IBM AIX System Requirements

Component	CPU	RAM	Disk Space
Repository	450 MHz	256 MB	900 MB
Logical Host	450 MHz	512 MB	450 MB

Table 18 Red Hat and SuSE Linux System Requirements

Component	CPU	RAM	Disk Space
Repository	1.2 GHz	256 MB	900 MB
Logical Host	1.2 GHz	512 MB	350 MB

Table 19 Sun Solaris System Requirements

Component	CPU	RAM	Disk Space
Repository	400 MHz	256 MB	850 MB
Logical Host	400 MHz	512 MB	400 MB

Table 20 HP NonStop Server System Requirements

Component	Server	RAM	Disk Space
Repository	HP NonStop Server, S Series	256 MB	512 MB
Logical Host	HP NonStop Server, S Series	512 MB	512 MB

Table 21 IBM z/OS System Requirements

Component	RAM	Disk Space
Logical Host	512 MB	350 MB

The *SeeBeyond ICAN Suite Installation Guide* contains additional information about the system requirements. The **Readme.txt** file (located in the root directory of the Repository CD-ROM) contains the most up-to-date system requirements.

Additional Sun Solaris Requirements

If you want to run a SeeBeyond Integration Server on Sun Solaris in 64-bit mode, then you must do the following:

- Use Sun Solaris 9.
- Add the **-d64** argument to the **JVM Args** property of the SeeBeyond Integration Server. To access the **JVM Args** property, start Enterprise Designer, display the Environment Explorer, right-click the Integration Server, and click **Properties**.

Important: You cannot run the SeeBeyond Integration Server on Sun Solaris 8 in 64-bit mode.

Configuring Failover Support in a Windows Clustering Environment

This appendix describes how to configure failover support for the Repository and the Logical Host using Microsoft Windows 2000 and Windows Server 2003 clustering technologies.

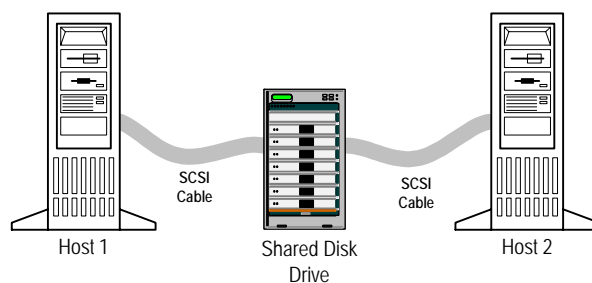
What's in This Chapter

- “Windows Clustering Overview” on page 63
- “Configuring Support for Repository Failover” on page 64
- “Configuring Support for Logical Host Failover” on page 73

7.1 Windows Clustering Overview

A *cluster* consists of two or more independent computers that work together as one system. These computers are referred to as *nodes*. Each node includes an instance of the Cluster service. Each node is connected to a shared storage device. Figure 14 shows a cluster with two nodes.

Figure 14 Cluster With Two Nodes



A group is a collection of resources that are assigned to a node. Examples of resources include physical disks, IP addresses, and network names.

A node can be active or passive. You can set up a cluster in which one node will be active and the other node(s) will be passive. This configuration is called *active/passive*. You also can set up a cluster in which all of the nodes are active. This configuration is called *active/active*.

7.2 Configuring Support for Repository Failover

This section describes how to configure support for Repository failover in a Windows clustering environment. The instructions are divided into multiple subsections. Perform the instructions in the order shown.

7.2.1 Overview

In the Repository failover environment, one node will be active and the other node(s) will be passive. This configuration is called *active/passive*.

During the configuration process, you use Microsoft's Cluster Administrator tool to create a group that contains the following resources:

- Shared disk drive
- IP address
- Network name
- The Repository itself

In addition, you install the Repository on the shared disk drive.

The active node is the owner of the group. If one of the resources fails, the group is automatically moved to another node and started on that node. This process is called *failover*. The new node becomes the active node.

The Repository is not cluster aware. When a failover is completed, the Repository does not remember the state that existed before the failover. This limitation could affect the users. For example:

- If an Enterprise Designer user is saving objects to the Repository during the failover, then the save will not succeed. Once the failover is completed, the user will need to save the objects again. If an Enterprise Designer user is not saving objects during the failover, then the failover will be transparent to the user.
- When the failover begins, the browser session of all Enterprise Manager users will expire. Once the failover is completed, the users will need to login to Enterprise Manager again.

7.2.2 Requirements

Before you perform the following procedures, you must have the following:

- A functional active/passive cluster that consists of two or more nodes. The nodes must be running Windows 2000 Advanced Server, Windows 2000 Datacenter Server, or Windows Server 2003.

For detailed information about how to set up a cluster, see the Microsoft clustering documentation.

Note: *Microsoft has a certification program for hardware vendors that supply a "cluster system." When you go to a vendor to buy computers for a cluster, the vendor has to*

sell you two computers, set up in the specific way dictated by Microsoft. See the appropriate Microsoft documentation or Web page for details.

- A cluster shared drive that is available for the Repository installation. The shared drive must be mountable from each cluster node.
- The correct version of Cluster Administrator. Windows 2000 Advanced Server and Windows 2000 Datacenter Server were tested with Cluster Administrator version 5.0. Windows Server 2003 was tested with Cluster Administrator version 5.2.

7.2.3 Creating the ICAN Group

The first task is to create a cluster group in which you can include a set of resources.

Note: If you are using an existing group, you can skip this procedure.

To create the ICAN group

- 1 Start Cluster Administrator.
- 2 Select the cluster in which the Repository will run.
- 3 Right-click the **Groups** folder under the cluster, choose **New**, and then choose **Group**.

The **New Group** wizard appears.

- 4 Enter a name for the group (for example, **ICAN Group**). Enter a description for the group. Click **Next**.

The **Preferred Owners** dialog box appears.

- 5 Move the nodes to the **Preferred owners** list and arrange them in order of preference. Click **Finish**.

7.2.4 Creating a Physical Disk Resource

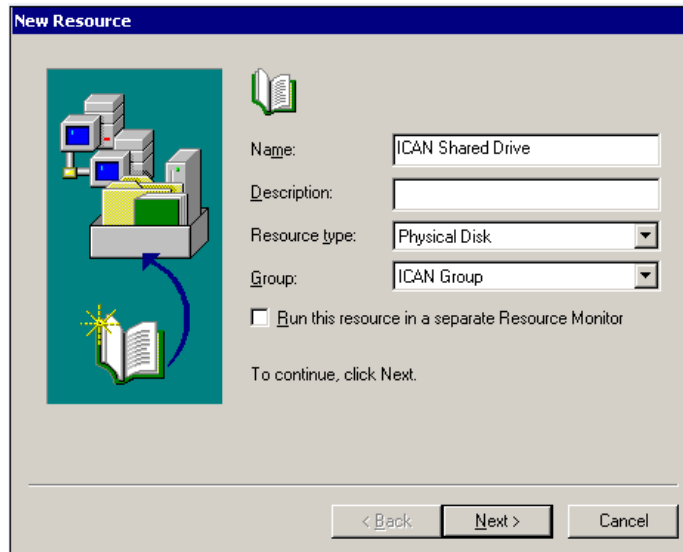
You must now create a Physical Disk resource within the ICAN group. From the Cluster Administrator, perform the following steps.

To create a Physical Disk resource

- 1 Right-click the ICAN group, choose **New**, and then choose **Resource**.

The **New Resource** dialog box appears.

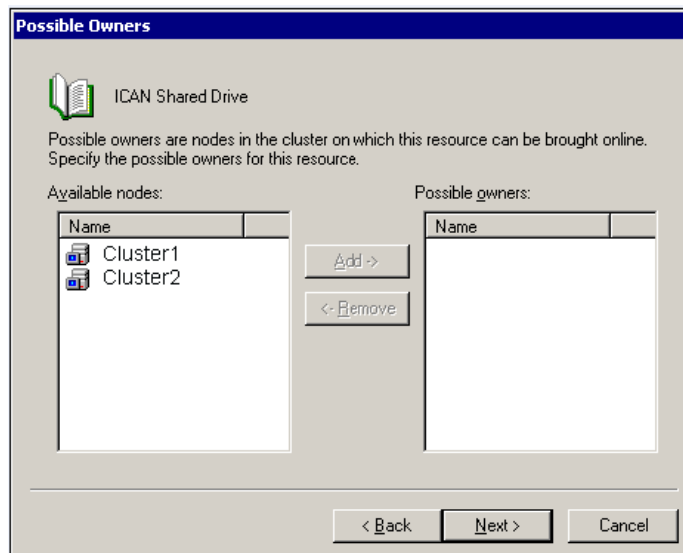
Figure 15 New Resource Dialog Box



- 2 Enter a name for the resource (for example, **ICAN Shared Drive**). Enter a description for the resource. Set the resource type to **Physical Disk**. Set the group to the ICAN group. When finished, click **Next**.

The **Possible Owners** dialog box appears.

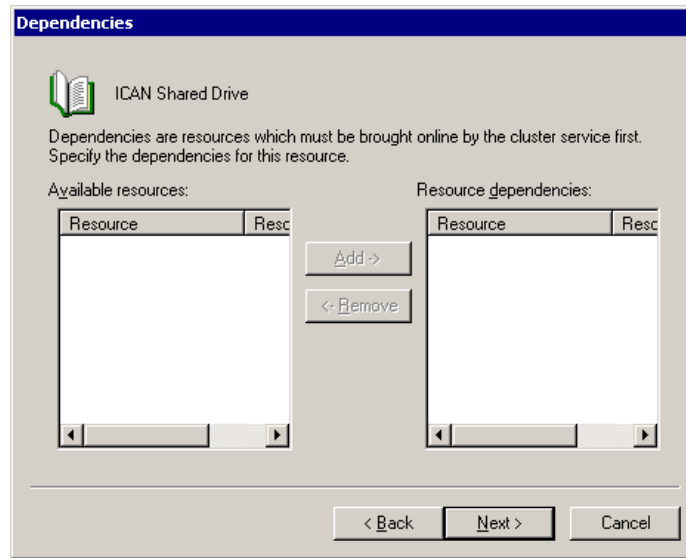
Figure 16 Possible Owners Dialog Box



- 3 Move the nodes that will be part of the cluster to the **Possible owners** list. Click **Next**.

The **Dependencies** dialog box appears.

Figure 17 Dependencies Dialog Box



- 4 The Physical Disk resource is not dependent on other resources. Therefore, click **Next**.
The **Disk Parameters** dialog box appears.
- 5 Choose the shared disk. Click **Finish**.
- 6 To ensure that the Physical Disk resource is accessible from each node, move the ICAN group into each node's Active Groups folder and verify that the group can be brought online at each node.

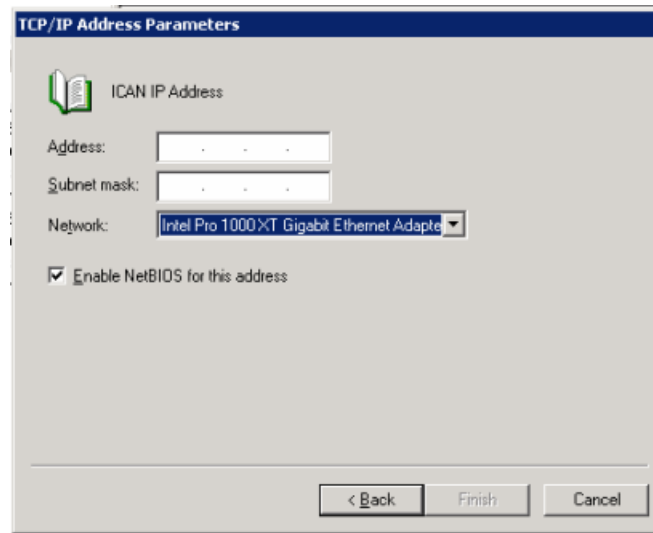
7.2.5 Creating an IP Address Resource

You must now create an IP Address resource within the ICAN group. From the Cluster Administrator, perform the following steps.

To create an IP Address resource

- 1 Right-click the ICAN group, choose **New**, and then choose **Resource**.
The **New Resource** dialog box appears.
- 2 Enter a name for the resource (for example, **ICAN IP Address**). Enter a description for the resource. Set the resource type to **IP Address**. Set the group to the ICAN group. When finished, click **Next**.
The **Possible Owners** dialog box appears.
- 3 Move the nodes that will be part of the cluster to the **Possible owners** list. Click **Next**.
The **Dependencies** dialog box appears.
- 4 The IP Address resource is not dependent on other resources. Therefore, click **Next**.
The **TCP/IP Address Parameters** dialog box appears.

Figure 18 TCP/IP Address Parameters Dialog Box



- 5 Enter the IP address and subnet mask of the cluster. Choose the network card. Click **Finish**.
- 6 To ensure that the IP Address resource is accessible from each node, move the ICAN group into each node's Active Groups folder and verify that the group can be brought online at each node.

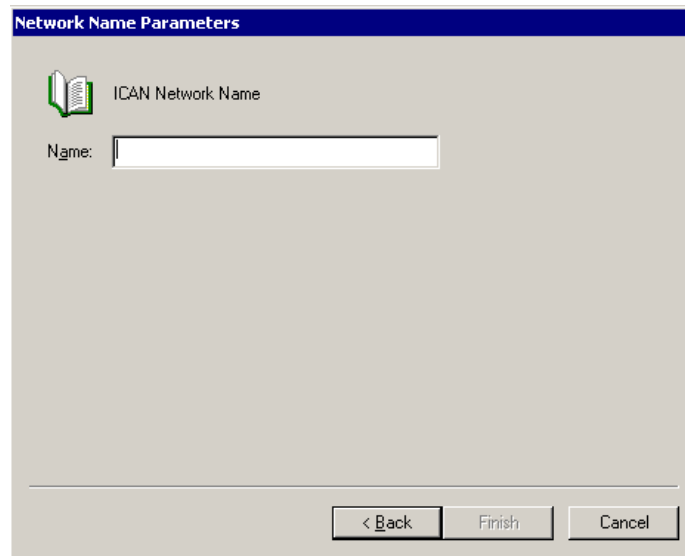
7.2.6 Creating a Network Name Resource

You must now create a Network Name resource within the ICAN group. From the Cluster Administrator, perform the following steps.

To create a Network Name resource

- 1 Right-click the ICAN group, choose **New**, and then choose **Resource**.
The **New Resource** dialog box appears.
- 2 Enter a name for the resource (for example, **ICAN Network Name**). Enter a description for the resource. Set the resource type to **Network Name**. Set the group to the ICAN group. When finished, click **Next**.
The **Possible Owners** dialog box appears.
- 3 Move the nodes that will be part of the cluster to the **Possible owners** list. Click **Next**.
The **Dependencies** dialog box appears.
- 4 The Network Name resource is dependent on the IP Address resource. Therefore, move the IP Address resource to the **Resource dependencies** list. When finished, click **Next**.
The **Network Name Parameters** dialog box appears.

Figure 19 Network Name Parameters Dialog Box



- 5 Enter the network name of the cluster. Click **Finish**.
- 6 To ensure that the Network Name resource is accessible from each node, move the ICAN group into each node's Active Groups folder and verify that the group can be brought online at each node.

7.2.7 Installing the Repository as a Windows Service

Install the Repository on one of the cluster nodes where the shared disk drive is accessible. During the installation process, you must do the following:

- Use a shared drive letter in the Repository location (for example, X:\ican50).
- Specify that you want to run the Repository as a Windows service. If you are using the GUI installation program, do this by selecting the **Run repository as Windows Service** check box in the **Repository Configuration** dialog box. If you are using the command line prompts, do this by entering **y** at the applicable prompt.

For detailed instructions on installing the Repository, see the *SeeBeyond ICAN Suite Installation Guide*.

7.2.8 Duplicating the Registry Keys

When you installed the Repository as a Windows service in the previous subsection, a series of registry keys were created on that cluster node. You must copy these registry keys to the other cluster nodes.

There are two approaches to performing this task. SeeBeyond recommends that you use the first approach, because it is quicker and less prone to error.

Important: *In order to use the first approach, the drive letter must be the same on each cluster node. If the drive letters are not the same, then you must use the second approach.*

The steps for the first approach are:

- 1 From the Cluster Administrator, move the ICAN group to a node on which the Repository was *not* installed.
- 2 Run the **installwinsvc.bat** script in the `<shared_drive>:\<ican_home>\repository` directory on the node.
- 3 Repeat Step 1 and Step 2 for the other nodes (if any) on which the Repository was *not* installed.

The second approach involves specifying the registry keys when you create the Generic Service resource in the next subsection. You will need to add a large number of registry keys one at a time. The instructions are located in Step 6 of the next subsection.

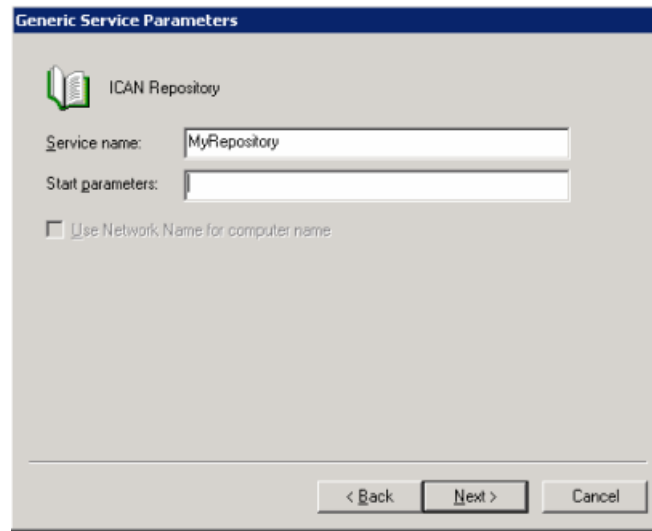
7.2.9 Creating a Generic Service Resource

You must now create a Generic Service resource within the ICAN group. This resource represents the Repository. From the Cluster Administrator, perform the following steps.

To create a Generic Service resource

- 1 Right-click the ICAN group, choose **New**, and then choose **Resource**.
The **New Resource** dialog box appears.
- 2 Enter a name for the resource (for example, **ICAN Repository**). Enter a description for the resource. Set the resource type to **Generic Service**. Set the group to the ICAN group. When finished, click **Next**.
The **Possible Owners** dialog box appears.
- 3 Move the nodes that will be part of the cluster to the **Possible owners** list. Click **Next**.
The **Dependencies** dialog box appears.
- 4 The Generic Service resource is dependent on the Physical Disk, IP Address, and Network Name resources. Therefore, move these resources to the **Resource dependencies** list. When finished, click **Next**.
The **Generic Service Parameters** dialog box appears.

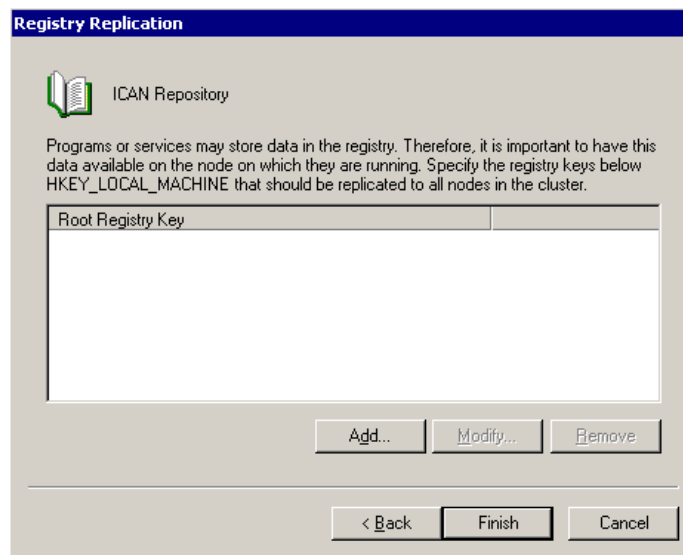
Figure 20 Generic Service Parameters Dialog Box



- 5 In the **Service name** field, enter the Repository name. Leave the **Start parameters** field blank. Click **Next**.

The **Registry Replication** dialog box appears.

Figure 21 Registry Replication Dialog Box



- 6 If you chose the second approach of duplicating the registry keys, add each of the following registry keys. These keys are located below **HKEY_LOCAL_MACHINE**. The exact names of the keys depend on the Repository name. The following list assumes that the Repository name is **MyRepository**. Note that for some of the keys, you must type the Repository name with all capital letters.

```
SYSTEM\CurrentControlSet\Enum\Root\LEGACY_MYREPOSITORY
SYSTEM\CurrentControlSet\Enum\Root\LEGACY_MYREPOSITORY\0000
SYSTEM\CurrentControlSet\Enum\Root\LEGACY_MYREPOSITORY\0000\Control
SYSTEM\CurrentControlSet\Services\Eventlog\Application\MyRepository
```

```
SYSTEM\CurrentControlSet\Services\MyRepository
SYSTEM\CurrentControlSet\Services\MyRepository\Enum
SYSTEM\CurrentControlSet\Services\MyRepository\Parameters
SYSTEM\CurrentControlSet\Services\MyRepository\Security

SYSTEM\ControlSet001\Enum\Root\LEGACY_MYREPOSITORY
SYSTEM\ControlSet001\Enum\Root\LEGACY_MYREPOSITORY\0000
SYSTEM\ControlSet001\Enum\Root\LEGACY_MYREPOSITORY\0000\Control
SYSTEM\ControlSet001\Services\Eventlog\Application\MyRepository
SYSTEM\ControlSet001\Services\MyRepository
SYSTEM\ControlSet001\Services\MyRepository\Enum
SYSTEM\ControlSet001\Services\MyRepository\Parameters
SYSTEM\ControlSet001\Services\MyRepository\Security

SYSTEM\ControlSet002\Enum\Root\LEGACY_MYREPOSITORY
SYSTEM\ControlSet002\Enum\Root\LEGACY_MYREPOSITORY\0000
SYSTEM\ControlSet002\Services\Eventlog\Application\MyRepository
SYSTEM\ControlSet002\Services\MyRepository
SYSTEM\ControlSet002\Services\MyRepository\Parameters
SYSTEM\ControlSet002\Services\MyRepository\Security
```

7 Click **Finish**.

- 8 To ensure that the Generic Service resource is accessible from each node, move the ICAN group into each node's Active Groups folder and verify that the group can be brought online at each node.

7.2.10 Using the Repository in a Windows Clustering Environment

In a non-Windows clustering environment, you start and stop the Repository by running a script.

In a Windows clustering environment, you start and stop the Repository from Cluster Administrator. Right-click the Generic Service resource (which represents the Repository) and choose **Bring Online** or **Take Offline**, respectively.

Note: The context menu also contains an item called **Initiate Failure**. You can choose this item for testing purposes to simulate a failover.

When the Repository is running, you can connect to it from Enterprise Designer and Enterprise Manager. In the hostname portion of the URL, use the *cluster name* rather than a node name. You will automatically be connected to the active node.

For example, assume that:

- You have a two-node cluster.
- The cluster name is **MyWindows2003Cluster.acme.com**.
- The name of node 1 is **MyWindows2003Node1.acme.com**.
- The name of node 2 is **MyWindows2003Node2.acme.com**.
- The base port number of the Repository is 12000.

To access Enterprise Designer or Enterprise Manager, you would use **http://MyWindows2003Cluster.acme.com:12000**.

7.3 Configuring Support for Logical Host Failover

This section describes how to configure support for Logical Host failover in a Windows clustering environment. The instructions are divided into multiple subsections. Perform the instructions in the order shown.

7.3.1 Overview

In the Logical Host failover environment, one node will be active and the other node(s) will be passive. This configuration is called *active/passive*.

During the configuration process, you use Microsoft's Cluster Administrator tool to create a group that contains the following resources:

- Shared disk drive
- IP address
- Network name
- The Logical Host itself

In addition, you install the Logical Host on the shared disk drive.

The active node is the owner of the group. If one of the resources fails, the group is automatically moved to another node and started on that node. This process is called *failover*. The new node becomes the active node.

7.3.2 Requirements

Before you perform the following procedures, you must have the following:

- A functional active/passive cluster that consists of two or more nodes. The nodes must be running Windows 2000 Advanced Server, Windows 2000 Datacenter Server, or Windows Server 2003.

For detailed information about how to set up a cluster, see the Microsoft clustering documentation.

Note: *Microsoft has a certification program for hardware vendors that supply a "cluster system." When you go to a vendor to buy computers for a cluster, the vendor has to sell you two computers, set up in the specific way dictated by Microsoft. See the appropriate Microsoft documentation or Web page for details.*

- A cluster shared drive that is available for the Logical Host installation. The shared drive must be mountable from each cluster node.
- The correct version of Cluster Administrator. Windows 2000 Advanced Server and Windows 2000 Datacenter Server were tested with Cluster Administrator version 5.0. Windows Server 2003 was tested with Cluster Administrator version 5.2.

7.3.3 Creating the ICAN Group

The first task is to create a cluster group in which you can include a set of resources.

Note: If you are using an existing group, you can skip this procedure.

To create the ICAN group

- 1 Start Cluster Administrator.
- 2 Select the cluster in which the Logical Host will run.
- 3 Right-click the **Groups** folder under the cluster, choose **New**, and then choose **Group**.

The **New Group** wizard appears.

- 4 Enter a name for the group (for example, **ICAN LH Group**). Enter a description for the group. Click **Next**.

The **Preferred Owners** dialog box appears.

- 5 Move the nodes to the **Preferred owners** list and arrange them in order of preference. Click **Finish**.

7.3.4 Creating a Physical Disk Resource

You must now create a Physical Disk resource within the ICAN group. From the Cluster Administrator, perform the following steps.

To create a Physical Disk resource

- 1 Right-click the ICAN group, choose **New**, and then choose **Resource**.

The **New Resource** dialog box appears.

- 2 Enter a name for the resource (for example, **ICAN LH Shared Drive**). Enter a description for the resource. Set the resource type to **Physical Disk**. Set the group to the ICAN group. When finished, click **Next**.

The **Possible Owners** dialog box appears.

- 3 Move the nodes that will be part of the cluster to the **Possible owners** list. Click **Next**.

The **Dependencies** dialog box appears.

- 4 The Physical Disk resource is not dependent on other resources. Therefore, click **Next**.

The **Disk Parameters** dialog box appears.

- 5 Choose the shared disk. Click **Finish**.

- 6 To ensure that the Physical Disk resource is accessible from each node, move the ICAN group into each node's Active Groups folder and verify that the group can be brought online at each node.

7.3.5 Creating an IP Address Resource

You must now create an IP Address resource within the ICAN group. From the Cluster Administrator, perform the following steps.

To create an IP Address resource

- 1 Right-click the ICAN group, choose **New**, and then choose **Resource**.
The **New Resource** dialog box appears.
- 2 Enter a name for the resource (for example, **ICAN LH IP Address**). Enter a description for the resource. Set the resource type to **IP Address**. Set the group to the ICAN group. When finished, click **Next**.
The **Possible Owners** dialog box appears.
- 3 Move the nodes that will be part of the cluster to the **Possible owners** list. Click **Next**.
The **Dependencies** dialog box appears.
- 4 The IP Address resource is not dependent on other resources. Therefore, click **Next**.
The **TCP/IP Address Parameters** dialog box appears.
- 5 Enter the IP address and subnet mask of the cluster. Choose the network card. Click **Finish**.
- 6 To ensure that the IP Address resource is accessible from each node, move the ICAN group into each node's Active Groups folder and verify that the group can be brought online at each node.

7.3.6 Creating a Network Name Resource

You must now create a Network Name resource within the ICAN group. From the Cluster Administrator, perform the following steps.

To create a Network Name resource

- 1 Right-click the ICAN group, choose **New**, and then choose **Resource**.
The **New Resource** dialog box appears.
- 2 Enter a name for the resource (for example, **ICAN LH Network Name**). Enter a description for the resource. Set the resource type to **Network Name**. Set the group to the ICAN group. When finished, click **Next**.
The **Possible Owners** dialog box appears.
- 3 Move the nodes that will be part of the cluster to the **Possible owners** list. Click **Next**.
The **Dependencies** dialog box appears.
- 4 The Network Name resource is dependent on the IP Address resource. Therefore, move the IP Address resource to the **Resource dependencies** list. When finished, click **Next**.
The **Network Name Parameters** dialog box appears.

- 5 Enter the network name of the cluster. Click **Finish**.
- 6 To ensure that the Network Name resource is accessible from each node, move the ICAN group into each node's Active Groups folder and verify that the group can be brought online at each node.

7.3.7 Installing the Logical Host as a Windows Service

Install the Logical Host on one of the cluster nodes where the shared disk drive is accessible. During the installation process, you must do the following:

- Use a shared drive letter for the Logical Host location (for example, **X:\ican50**).
- Configure the Logical Host to run as a Windows service by running the **installwinsvc.bat** script in the **ICAN-root\logicalhost\bootstrap\bin** directory. You will need to enter the name of the Windows service in **“Creating a Generic Service Resource” on page 77**.

For detailed instructions on installing the Logical Host, see the *SeeBeyond ICAN Suite Installation Guide*.

For detailed instructions on configuring the Logical Host to run as a Windows service, see the *eGate Integrator System Administration Guide*.

7.3.8 Duplicating the Registry Keys

When you installed the Logical Host as a Windows service in the previous subsection, a series of registry keys were created on that cluster node. You must copy these registry keys to the other cluster nodes.

There are two approaches to performing this task. SeeBeyond recommends that you use the first approach, because it is quicker and less prone to error.

Important: *In order to use the first approach, the drive letter must be the same on each cluster node. If the drive letters are not the same, then you must use the second approach.*

The steps for the first approach are:

- 1 From the Cluster Administrator, move the ICAN group to a node on which the Logical Host was *not* installed.
- 2 Run the **installwinsvc.bat** script in the **<shared_drive>:\<ican_home>\logicalhost\bootstrap\bin** directory on the node.
- 3 Repeat Step 1 and Step 2 for the other nodes (if any) on which the Logical Host was *not* installed.

The second approach involves specifying the registry keys when you create the Generic Service resource in the next subsection. You will need to add a large number of registry keys one at a time. The instructions are located in Step 6 of the next subsection.

7.3.9 Creating a Generic Service Resource

You must now create a Generic Service resource within the ICAN group. This resource represents the Logical Host. From the Cluster Administrator, perform the following steps.

To create a Generic Service resource

- 1 Right-click the ICAN group, choose **New**, and then choose **Resource**.
The **New Resource** dialog box appears.
- 2 Enter a name for the resource (for example, **ICAN Logical Host**). Enter a description for the resource. Set the resource type to **Generic Service**. Set the group to the ICAN group. When finished, click **Next**.
The **Possible Owners** dialog box appears.
- 3 Move the nodes that will be part of the cluster to the **Possible owners** list. Click **Next**.
The **Dependencies** dialog box appears.
- 4 The Generic Service resource is dependent on the Physical Disk, IP Address, and Network Name resources. Therefore, move these resources to the **Resource dependencies** list. When finished, click **Next**.
The **Generic Service Parameters** dialog box appears.
- 5 In the **Service name** field, enter the name of the Logical Host Windows service. Leave the **Start parameters** field blank. Click **Next**.
The **Registry Replication** dialog box appears.
- 6 If you chose the second approach of duplicating the registry keys, add each of the following registry keys. These keys are located below **HKEY_LOCAL_MACHINE**. The exact names of the keys depend on the name of the Logical Host Windows service. The following list is based on the default name of **ICAN 5.0.5 LogicalHost**. Note that for some of the keys, you must type the name with all capital letters.

```

SYSTEM\CurrentControlSet\Enum\Root\LEGACY_ICAN_5.0.5_LOGICALHOST
SYSTEM\CurrentControlSet\Enum\Root\LEGACY_ICAN_5.0.5_LOGICALHOST\0000
SYSTEM\CurrentControlSet\Enum\Root\LEGACY_ICAN_5.0.5_LOGICALHOST\0000\Control
1
SYSTEM\CurrentControlSet\Services\Eventlog\Application\ICAN 5.0.5
LogicalHost
SYSTEM\CurrentControlSet\Services\ICAN 5.0.5 LogicalHost
SYSTEM\CurrentControlSet\Services\ICAN 5.0.5 LogicalHost\Enum
SYSTEM\CurrentControlSet\Services\ICAN 5.0.5 LogicalHost\Parameters
SYSTEM\CurrentControlSet\Services\ICAN 5.0.5 LogicalHost\Security

SYSTEM\ControlSet001\Enum\Root\LEGACY_ICAN_5.0.5_LOGICALHOST
SYSTEM\ControlSet001\Enum\Root\LEGACY_ICAN_5.0.5_LOGICALHOST\0000
SYSTEM\ControlSet001\Enum\Root\LEGACY_ICAN_5.0.5_LOGICALHOST\0000\Control
SYSTEM\ControlSet001\Services\Eventlog\Application\ICAN 5.0.5 LogicalHost
SYSTEM\ControlSet001\Services\ICAN 5.0.5 LogicalHost
SYSTEM\ControlSet001\Services\ICAN 5.0.5 LogicalHost\Enum
SYSTEM\ControlSet001\Services\ICAN 5.0.5 LogicalHost\Parameters
SYSTEM\ControlSet001\Services\ICAN 5.0.5 LogicalHost\Security

SYSTEM\ControlSet002\Enum\Root\LEGACY_ICAN_5.0.5_LOGICALHOST
SYSTEM\ControlSet002\Enum\Root\LEGACY_ICAN_5.0.5_LOGICALHOST\0000
SYSTEM\ControlSet002\Services\Eventlog\Application\ICAN 5.0.5 LogicalHost
SYSTEM\ControlSet002\Services\ICAN 5.0.5 LogicalHost

```

```
SYSTEM\ControlSet002\Services\ICAN 5.0.5 LogicalHost\Parameters  
SYSTEM\ControlSet002\Services\ICAN 5.0.5 LogicalHost\Security
```

7 Click **Finish**.

8 To ensure that the Generic Service resource is accessible from each node, move the ICAN group into each node's Active Groups folder and verify that the group can be brought online at each node.

7.3.10 Using the Logical Host in a Windows Clustering Environment

In a non-Windows clustering environment, you start the Logical Host by running a script, and you stop the Logical Host either from the ICAN Monitor or by running a script.

In a Windows clustering environment, you start and stop the Logical Host from Cluster Administrator. Right-click the Generic Service resource (which represents the Logical Host) and choose **Bring Online** or **Take Offline**, respectively.

***Note:** The context menu also contains an item called **Initiate Failure**. You can choose this item for testing purposes to simulate a failover.*

One of the Logical Host properties is the physical host name. When you specify this property in the **logical-host.properties** file, you must use the network name of the cluster, rather than the network name of a cluster node.

Case Study

This chapter uses a Web Order scenario to illustrate the various phases of ICAN Suite deployment.

What's in This Chapter

- [“Requirements Analysis and Deployment Planning” on page 79](#)
- [“System Design and Development” on page 80](#)
- [“Testing, Transition to Production, and Maintenance” on page 82](#)

8.1 Requirements Analysis and Deployment Planning

A company wants to create a composite application that orchestrates the flow of activities in the order fulfillment process. The participants in this process are:

- Customers
- Inventory department
- Shipping department
- Accounting department

The deployment project team includes a project manager, a system administrator, a business analyst, two developers, and a quality assurance (QA) engineer.

At the kick-off meeting, the project team decides to have status meetings once a week.

The business analyst gathers information about the workflow. The following steps represent a high-level view of the workflow:

- 1 A customer places an order through the company's Web order system.
- 2 The inventory system is queried to check the availability of the items.
- 3 If the items are available, the following actions are performed.
 - A A shipping request is sent to the shipping department.
 - B A billing request is sent to the accounting department.
 - C The shipping department mails the items and an invoice to the customer.
 - D The shipping department sends an e-mail to the customer indicating that the items have been sent.

- 4 If the items are not available, the Web order system informs the customer of this situation.

The workflow handles various types of exceptional conditions. For example, the accounting system may be temporarily unavailable.

The project team determines that the following ICAN Suite products will be used:

- eGate Integrator
- eInsight Business Process Manager
- HTTP eWay
- eMail eWay

The system administrator installs the Repository on a UNIX system, uploads the appropriate products from the ICAN Suite CD-ROMs to the Repository, and downloads Enterprise Designer from the Repository to his Windows computer. The administrator then creates Repository users for the developers and the QA engineer.

The developers and the QA engineer download Enterprise Designer from the Repository to their Windows computers.

The QA engineer writes the initial test plan.

8.2 System Design and Development

Working in Enterprise Designer, the development team creates a new Project called **prjWebOrder**.

The development team adds a Business Process called **bpWebOrder** to the Project. The Business Process is based on the workflow that was identified during the requirements analysis and deployment planning phases.

The first activity in the Business Process is a **Receive** activity that performs a blocking wait. When an order arrives, the activity fetches the order from the Web server. See Figure 22.

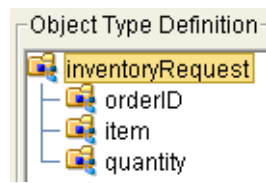
Figure 22 Receive Activity in Business Process



The different systems in the workflow use different formats to represent a customer order. The development team creates Java-based Collaborations to transform the data between these systems.

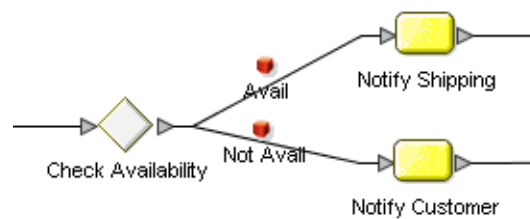
As part of this task, the development team creates a number of User-Defined Object Type Definitions (OTDs). See Figure 23 for an example.

Figure 23 User-Defined Object Type Definition for Inventory Request



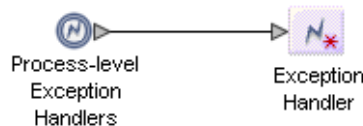
The Business Process contains various points at which one of several possible paths can be taken. The development team uses the **Decision** branching activity for these situations. For example, the Business Process checks for the availability of the items that the customer ordered. If the items are available, a series of actions take place. If the items are not available, a different series of actions take place. See Figure 24.

Figure 24 Decision Branching Activity in Business Process



The development team uses various methods to handle the exceptional conditions that the business analyst identified. Figure 25 shows a **Catch All Exceptions** element.

Figure 25 Catch All Exceptions Element in Business Process



To run a Project, you must create an Environment and a Deployment Profile. The Deployment Profile specifies how the Project components are deployed in the Environment.

The project team decides to create separate versions of these components for the development, test, and production phases. The development version of the Environment is called **envWebOrderDev**. The development version of the Deployment Profile is called **dpWebOrderDev**.

During the system design and development phase, the QA engineer refines the initial test plan.

8.3 Testing, Transition to Production, and Maintenance

eGate contains a tool that checks the validity of Java-based Collaborations. eInsight contains a tool that checks for errors in the Business Process Model. The developers use these tools during unit testing.

Once the unit testing is finished, the QA engineer begins the integration testing.

The test version of the Environment is called **envWebOrderTest**. The test version of the Deployment Profile is called **dpWebOrderTest**.

To test the error handling, the QA engineer sends various types of invalid data through the Business Process. The QA engineer discovers that the Business Process does not properly handle ZIP codes with alphabetic characters, so the development team adds the appropriate error handling.

To test whether the system can handle the expected load, the QA engineer uses a third-party tool to simulate customer orders.

During the transition-to-production phase, the project team creates new versions of the Environment and Deployment Profile:

- The production version of the Environment is called **envWebOrderPrd**.
- The production version of the Deployment Profile is called **dpWebOrderPrd**.

Maintenance personnel use the ICAN Monitor and the Alert Agent to check whether the system is running correctly.

eGate Integrator on HP NonStop Server

This appendix provides information about deploying and maintaining eGate Integrator on an HP NonStop Server system.

What's in This Appendix

- [“Introduction to eGate on HP NonStop Server” on page 83](#)
- [“Installation and Configuration of eGate Integrator on HP NonStop Server” on page 84](#)
- [“LHInit Program” on page 84](#)
- [“Repository and Pathway” on page 85](#)
- [“Transaction Monitoring Facility” on page 86](#)
- [“Configuring for Maximum Scalability” on page 86](#)
- [“Configuring Logical Hosts to use Parallel Library TCP/IP” on page 87](#)
- [“Troubleshooting” on page 87](#)

A.1 Introduction to eGate on HP NonStop Server

eGate Integrator provides a number of features, and takes advantage of many of HP NonStop Server's native facilities, to support scalability, reliability, and fault tolerance, when deployed on an HP NonStop Server system.

Scalability

Scalability, ensuring that the eGate Integrator system supports increased workloads and heavy demand periods, is achieved by utilizing multiple CPUs.

eGate uses message queuing, as configured by the user. All eGate components configured to consume from queues, do so in a round-robin sequence. This round-robin approach spreads the load across available CPUs.

The round-robin sequence is managed by HP NonStop Server JMS. As soon as a Collaboration on a Logical Host CPU is available, it requests to consume from the HP NonStop Server JMS queue.

Reliability

If a Logical Host's CPU fails, the generic process LHInit will terminate, along with the rest of the Logical Host components on the failed CPU. When the CPU is restored, the HP NonStop Server starts the installed SeeBeyond LHInit process. LHInit then starts the rest of the eGate components in the Logical Host running in that CPU.

In addition, the SeeBeyond-supplied script **pathway.sh** invokes the capabilities of **pathmon** (native to HP NonStop Server) to provide high availability failover for eGate Repository processes.

Fault Tolerance

When a Collaboration is configured to use XA transactions, any transaction in process by a Collaboration on a CPU that fails, that is not committed at the time of the failure, is rolled back and retained in the queue.

In the event of a CPU failure, the next available Collaboration running on an available configured eGate Logical Host CPU handles the queued transaction until the CPU that went down is restored or a new CPU is installed.

Note: SeeBeyond recommends that you always use XA transactions in Collaborations for improved recoverability and once-only message delivery.

A.2 Installation and Configuration of eGate Integrator on HP NonStop Server

For directions on the installation and configuration of eGate Integrator on an HP NonStop Server system, including system requirements, Repository installation, and Logical Host installation, see the *SeeBeyond ICAN Suite Installation Guide*.

A.3 LHInit Program

The LHInit program starts the Logical Host and restarts eGate in the event of a CPU failure. When the Logical Host is first loaded onto HP NonStop Server, LHInit must be manually installed and started as a generic process. It is made persistent by HP NonStop Server system software.

If a CPU failure occurs under normal operation, the LHInit generic process is terminated along with the rest of the Logical Host components on the failed CPU. Upon restart of the failed CPU, the HP NonStop Server starts the LHInit process, which starts eGate. Once all of the eGate components have started, the Logical Host is ready to handle actions related to the operation of the eGate components.

LHInit must be installed on each CPU that is running a Logical Host. Each Logical Host, running on the same computer under the same TCP/IP process, must be configured to use different ports which are not already in use.

The *SeeBeyond ICAN Suite Installation Guide* describes how to perform the following tasks:

- Installing the Logical Host
- Modifying the Logical Host startup configuration file
- Running the LHInit program

A.4 Repository and Pathway

The Repository stores the Project (logical components and configuration) and Environment (physical resources and configuration) information required for an implementation. The Repository installation includes the SeeBeyond Repository Server and the Repository Library. The Repository Server manages the SeeBeyond Repository, the Enterprise Manager, as well as SNMP processes.

A Windows computer with FTP capabilities is required to install the Repository on HP NonStop Server. For directions on installing the Repository on HP NonStop Server, see the *SeeBeyond ICAN Suite Installation Guide*.

A.4.1. Repository Under Pathway for HP NonStop Server

The **pathmon** facility (native to HP NonStop Server) monitors processes to protect them against CPU failure. The SeeBeyond-supplied script **pathway.sh** invokes some of the **pathmon** capabilities to provide high availability failover for eGate Repository processes.

Pathway allows the user to:

- Check the status of a Repository pathway process
- Stop execution of a running Repository pathway process
- Resume execution of a frozen Repository pathway process
- Permanently delete a Repository pathway process

The pathway process automatically restarts in case a JVM failure or bad user code causes the Repository Server to terminate abnormally or in case of a CPU failure. In the case of a restart, users will be able to reconnect to the Repository after restarting the Enterprise Designer—the Enterprise Designer must be restarted to obtain a new socket.

For directions on running Repository Pathway process and Pathway commands, see the *SeeBeyond ICAN Suite Installation Guide*.

A.5 Transaction Monitoring Facility

The HP NonStop Server provides Transaction Monitoring Facility services with the HP NonStop Transaction Manager/Massively Parallel (NonStop TM/MP) service. This service protects a database from damage caused by transaction failure, operator error, and file or volume loss. It does so by recording recovery data in an audit trail file. This data allows the NonStop TM/MP service to rebuild the database after a failure occurs.

The SeeBeyond Message Server utilizes NonStop TM/MP to provide XA services. Collaborations configured to use XA will take advantage of the NonStop TM/MP services. Alternatively, users can use JTA User Transactions within their Collaborations to utilize the NonStop TM/MP services.

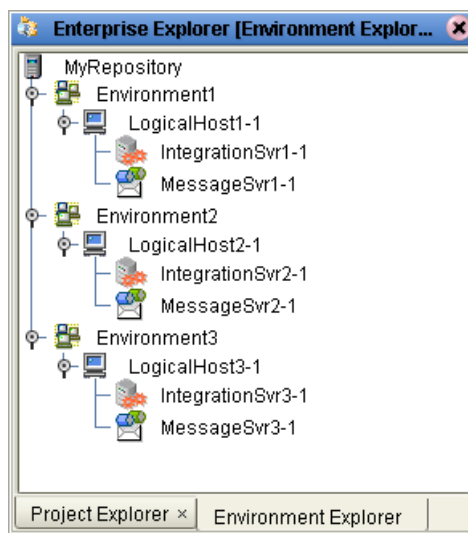
The advantage to using XA is that the messages are protected by NonStop TM/MP as they enter and leave the Collaboration. Collaborations using JTA User Transactions only utilize NonStop TM/MP while the messages are being processed within the Collaboration.

A.6 Configuring for Maximum Scalability

When eGate is deployed on HP NonStop Server, you can increase throughput by utilizing multiple CPUs. This method would be appropriate when each of the deployed Collaborations is intended to execute the same business logic.

First, add a unique Environment, Logical Host, Integration Server, and Message Server for each CPU that you want to use (see Figure 26). Then create a unique Deployment Profile for each Environment and deploy the Collaboration to each of the Integration Servers.

Figure 26 Utilizing Multiple CPUs



When creating multiple Environments, be sure that the base port numbers of the Logical Hosts do not conflict with each other. The “Environments” chapter in the *eGate Integrator User’s Guide* describes how to configure the Logical Host base port numbers.

Note: *The Logical Host uses a range of 10 ports starting with the base port number, so you must take this into consideration. For example, you could set the base port number for LogicalHost1 to 18000, the base port number for LogicalHost2 to 18100, and the base port number for LogicalHost3 to 18200.*

A.7 Configuring Logical Hosts to use Parallel Library TCP/IP

Multiple Logical Hosts can be configured to share TCP/IP listening ports through the use of the HP NonStop Server’s Parallel Library TCP/IP functionality. Parallel Library TCP/IP provides improved scalability, improved availability, and simplified configuration.

For more information, see the *TCP/IP eWay Intelligent Adapter User’s Guide*.

A.8 Troubleshooting

This section describes how to troubleshoot a couple of problems that you may encounter.

A.8.1. Signal 31 Message

If the HP NonStop Server returns a signal 31 message when you start the Logical Host, then you need to reduce the size of the QIO segment. The QIO segment is stored in the user addressable data space.

For each CPU, do the following:

- Set the segment size to 128 MB.
- Set the base address to 0x38000000.

The *QIO Configuration and Management Manual* provided with your HP NonStop Server describes how to configure these settings.

A.8.2. Repository Start Time Is Slow

If the Repository is taking a long time to start, check the size of the **EventRepositoryDb.script** file in the *ican_home/repository/EventRepositoryDb* directory. If the file has grown too large, perform the following steps to empty the Event Repository database:

- 1 Rename the file by adding the date at the end. For example:

```
mv EventRepository.script EventRepository.script.2003-12-16
```

- 2 Go to the **eManager\monitor\bin** directory on the Repository installation CD-ROM and copy the **EventRepositoryDb.script** file to the *ican_home/repository/EventRepositoryDb* directory on your HP NonStop Server.
- 3 Restart the Repository.

Glossary

BI

Business integration (also Business Intelligence).

Collaboration

A logical operation performed between some combination of message destinations and external applications. The operation is defined by a Collaboration Definition, which can be encoded in either Java or XSLT.

Also see “**Service**” and “**Collaboration Definition**”.

Collaboration Definition

The encoding of business rules, in Java or XSLT format. Typically, the encoding consists of operations on OTDs (see “**OTD**”). Several Collaborations can have the same Collaboration Definition.

Connection

Consists of the configuration information that enables an eWay to connect to an external system.

Connectivity Map

Contains business logic and routing information about the data transmission. A Connectivity Map usually includes one or more Collaborations, Passthrough Collaborations, topics, queues, and eWays. A Connectivity Map is created under a Project. A Project may have multiple Connectivity Maps.

Constants

A name or value pair that is visible across a Project.

CRM

Customer Relations Management

Data Cleansing

Data must be cleansed of errors in structure and content before it is useful in data warehousing and integration; this means transforming data for accurate and effective use in a database or data management system by cleansing “dirty” or redundant data.

Data Dictionary

Defines the organization of a database and lists all files in the database, the number of records in each file, and the names and types of each field. The data dictionary is often hidden from end users. Although the dictionary doesn’t contain actual data, it does contain essential information for managing the database.

Data Integrity

Refers to the accuracy and validity of data. Data integrity can be compromised in many ways, including human error through data entry, or through faulty logic in programming. Computer viruses, software bugs and many other factors can also compromise data integrity.

Data Mapping

In relational databases (RDBMSs) data mapping is the relationship and data flow between source and target objects. Mapping involves structuring the relationship between source and target objects.

Data Mart

A smaller, focused, database designed to help managers make business decisions. (A data warehouse is a larger, enterprise, database(s).)

Data Mining

Used to synthesize or isolate unique data patterns to predict future behaviors or to filter data to select patterns that help discover previously unknown relationships among data. Commonly used by marketers who acquire and distill consumer information.

Data Transformation

Data transformation is necessary after extracting data from legacy data formats, or any format that requires cleansing. Data is transformed for efficient use for Business-to-Business Enterprise Data Integration.

Data Warehouse

A copy or view of enterprise transaction data (sometimes non-transaction data) that is used for reporting. The data is often summarized and always structured for queries and analysis.

Deployment Profile

Contains the information about how the Project components will be deployed in an Environment. A Project can have multiple Deployment Profiles, but only one Deployment Profile can be activated for a Project in any one Environment.

Derived Collaboration

Collaboration that inherits operations from another, according to standard object-oriented practice.

Dimension Table

Dimension tables describe the business entities of an enterprise; also called lookup or reference tables.

Dirty Data

Dirty data contains, but is not limited to, incorrect data including spelling errors, punctuation errors, incorrect data referencing, incomplete, inconsistent, outdated, and redundant data.

Drill Down

To move from summary to more detailed data by “drilling down” to get it. In database terminology this might mean starting with a general category and drilling down to a specific field in a record.

eGate System

See “Project”.

Environment

A collection of physical resources and their configurations that are used to host Project components. An Environment contains logical hosts and external systems.

EPR

Enterprise Resource Management

ETL

Extract, Transform, Load. Extract is the process of reading data from a source database and extracting the desired subset of data. Transform is the process of converting the extracted data from its previous form into the desired form. Load is the process of writing the data into a larger database.

eWay

A link between a Collaboration and an external connection including the message server connection (topic or queue) or external application.

External Application

A logical representation in an eGate Project of an external application.

External System

A representation in an eGate Project of an external application system.

Extraction

Data are extracted from a source using software tools. This first step in ETL initially “gets” the data.

Fact Table

A fact table typically contains two types of columns: those containing facts and those that contain foreign keys to dimension tables. Fact tables contain detail facts and/or summary facts.

ICAN Suite

The SeeBeyond Integrated Composite Application Network Suite.

Integration Server

J2EE software platform that houses the business logic container used to run Collaborations and JCA connectors (eWays). Provides transaction services, persistence, and external connectivity.

JMS IQ Manager

JMS-compliant, guaranteed delivery store, forwarding, and queueing service.

Join

Matches records, which are joined by a common field, in two tables in a relational database. Often part of a Select query.

Link

The JMS Connection between a Collaboration and a topic or queue in a JMS-compliant message server.

Linked Message Destination

A reference to a Message Destination defined in another Connectivity Map.

Logical Host

An instance of the eGate runtime Environment that is installed on a machine. A Logical Host contains the software and other installed components that are required at runtime, such as application and message servers.

Management Agent

Uses J2EE technology to manage and monitor an eGate 5.0 deployment that may contain other application servers in addition to the SeeBeyond Integration Server. Defines management interfaces and services designed for distributed environments, focusing on providing functionality for managing networks, systems, and applications.

Message Destination

A general term for a topic or queue. Two or more Projects can share a message destination that has the same name and is deployed on the same message server. A single Project may also have a single message destination referenced in multiple Connectivity Maps.

Metadata

“Data about data.” Metadata describes “how,” “when,” and “who” about structure and format, of a particular set of data. ETL tools are used to generate and maintain a central metadata repository.

Non-normalized Data

Non-normalized data cannot be cross-referenced accurately, if at all, and causes manageability issues. Non-normalized data may be converted to normalized data.

Normalized Data

Normalization is a common database design process used to remove redundant or incorrect organization and data. The design and normalization of the database will create a maintainable data set that can be cross-referenced.

Normalized data is not only easier to analyze but also easier to expand. Normalization involves removing redundancy and correcting incorrect data structure and organization.

OLAP

Online analytical processing.

OTD

An acronym for Object Type Definition. OTDs contain the data structure and rules that define an object. An OTD is used in Java Collaboration Definitions for creating data transformations and interfacing with external systems.

Project

Contains a collection of logical components, configurations, and files that are used to solve business problems. A Project organizes the files and packages and maintains the settings that comprise an eGate system in SeeBeyond's Enterprise Designer.

Query

A request for information from a database. There are three query methods:

Choose – With this easy-to-use method, the database system presents a list of parameters from which you can choose. This method is not as flexible as other methods.

Query by example (QBE) – With this method, the system lets you specify fields and values to define a query.

Query language – With this method, you have the flexibility and power to make requests for information in the form of a stylized query using a query language. This is the most complex and powerful method.

Queue

A JMS queue is a shareable object that conforms to the *point-to-point* (p2p, or PTP) messaging domain, where one sender delivers a message to exactly one receiver. When the SeeBeyond JMS IQ Manager sends a message to a queue, it ensures it is received once and only once, even though there may be many receivers "listening" to the queue. This is equivalent to the subscriber pooling in other queue implementations. You can reference a queue that exists in another Connectivity Map or Project.

Raw Data

Data that has not been turned into "information," through processing. Although factual and "real," raw data is unorganized.

Relational Database (RDBMS)

Short for Relational Database Management System, most often referred to as RDBMS. Data is stored in related tables. Relational databases can be viewed in many different ways.

In this system a single database can be spread across several tables. (RDBMS differs from flat-file databases where each database is self-contained as a single file or table.)

Repository

Stores and manages the setup, component, and configuration information for eGate Projects. The Repository also provides monitoring services for Projects, which include version control and impact analysis.

Schema Runtime Environment

An add-on in eGate 5.0 that provides the upgrade path for e*Gate 4.x users to upgrade to eGate 5.0. Also known as the SRE.

Service

Contains the information about executing a set of business rules. These business rules can be defined in a Java Collaboration Definition, XSLT Collaboration Definition, Business Process, eTL Definition, or other service. A Service also contains binding information for connecting to JMS Topics, Queues, eWays, and other services.

Staging Data

Data that is to be processed before entering the warehouse.

Subproject

An independent Project that is included as part of another Project and listed on the Enterprise Explorer tree beneath the main Project icon.

Table

Refers to data arranged in rows and columns, like a spreadsheet. In relational database management systems, all information is stored in tables.

Topic

A JMS topic is a shareable object that conforms to the *publish-and-subscribe* (pub/sub) messaging domain, where one publisher broadcasts messages to potentially many subscribers. When the SeeBeyond JMS IQ Manager publishes a message on a topic, it ensures that all subscribers receive the message.

Transformation

Data that are extracted from databases are transformed into a desired form, using various tools that cleanse, merge, purge, aggregate, calculate, audit, remove redundancy, standardize, etc.

XSLT

An acronym for Extensible Stylesheet Language Transformations. A file format used in eGate to generate Collaboration Definitions.

e*Gate 4.x Terms in eGate 5.0

Table 22 provides definitions for the terms that are new with eGate release 5.0, as well as equivalent terms from eGate release 4.x.

Table 22 eGate 5.0 Terms

5.0 Term	4.x Equivalent Term
Collaboration	Collaboration
Collaboration Definition	Collaboration Definition
Connection	eWay Connection
Connectivity Map	Closest: Network View of an entire Schema
Deploy	Run the Control Broker
Deployment	<none>
Deployment Profile	Closest: Schema
Enterprise Designer	Enterprise Manager
Enterprise Manager	Enterprise Monitor
Environment	Schema (except only includes physical information, not business logic)
eWay	eWay Connection eWay
eWay Configuration	eWay Connection Configuration
External Application	eWay Connection
External System	eWay Connection
JMS Connection	eWay Connection
Integration Server	<none>
Link	JMS eWay Connection
Linked Message Destination	<none>
Logical Host	Participating Host
Message Destination	Topic or queue
Message Server	JMS IQ Manager
Object Type Definition (OTD)	Event Type Definition (ETD)
Process Manager	Control Broker
Project	Schema (except not including physical layer)
Queue	JMS queue
Repository	Registry
Subproject	Schema

Table 22 eGate 5.0 Terms (Continued)

5.0 Term	4.x Equivalent Term
Topic	JMS topic
XSLT	<none>

Index

Numerics

- 64-bit mode
 - Sun Solaris requirement 62

A

- acceptance testing 27, 52
- Access Control Lists 42
- active/active cluster
 - defined 63
- active/passive cluster
 - defined 63
- Administrator user 39
- Alert Agent
 - troubleshooting with 53
- architecture 29
- availability 30

B

- bottlenecks 52
- business process management 16

C

- case study 79
- change management 28, 49, 55
- clustering *See* Windows clustering
- Composed Business Services layer 36
- composite application 79
- Composite Applications layer 36
- constraints 29
- conventions, document 11
- CPU requirements
 - HP Tru64 61
 - HP-UX 61
 - IBM AIX 61
 - Red Hat Linux 62
 - Sun Solaris 62
 - SuSE Linux 62
 - underutilization 45
 - Windows 61

D

- debugging 51
- deployment planning
 - documents 28
 - objectives 26
 - overview 21, 26
- Deployment Profile 42
- deployment project plan 28
- directory structure
 - multiple Logical Hosts 45
- disk space requirements
 - HP NonStop Server 62
 - HP Tru64 61
 - HP-UX 61
 - IBM AIX 61
 - IBM z/OS 62
 - Red Hat Linux 62
 - Sun Solaris 62
 - SuSE Linux 62
 - Windows 61
- document conventions 11
- documentation
 - related 12
 - standards 27

E

- eBAM Studio
 - overview 19
- eGate Integrator
 - overview 15
- eIndex
 - overview 19
- eInsight
 - error handling 44
 - overview 16
- Elemental Business Services layer 35
- Enterprise Designer 15
- Enterprise Manager 15
- Environment 44
- ePortal Composer
 - overview 18
- error handling 24, 44
- eTL Integrator
 - overview 18
- eView Studio
 - overview 19
- eVision Studio
 - error handling 44
 - overview 17
- eWay 29
- eXchange Integrator
 - overview 18

Existing Systems layer 35

F

failover

Logical Host 73

Repository 64

functional requirements specification 29

G

general model 26

H

hardware

determining requirements 57

HP NonStop Server

LHInit program 84

maximum scalability 86

Parallel Library TCP/IP 87

pathmon 85

pathway.sh 85

signal 31 87

system requirements

disk space 62

RAM 62

Transaction Monitoring Facility 86

troubleshooting 87

HP Tru64

system requirements

CPU 61

disk space 61

RAM 61

HP-UX

system requirements

CPU 61

disk space 61

RAM 61

I

IBM AIX

system requirements

CPU 61

disk space 61

RAM 61

IBM z/OS

system requirements

disk space 62

RAM 62

ICAN Monitor

troubleshooting with 53

installation, software and hardware 27

Integration Server

naming convention 41

scalability of 45

small Project recommendations 46

Sun Solaris requirement 62

integration testing 51

interfaces 26, 29, 30

isolating Projects 45

J

JMS Client

small Project recommendations 46

JVM Args property 62

L

location transparency 34

log files

troubleshooting with 53

logging

effect on performance 52

Logical Host

failover 73

multiple on single computer 45

naming convention 41

sharing 43

loose coupling 34

M

maintenance 54

milestones 29

N

naming conventions 40

O

Object Type Definition

naming convention 40

organization

test team 31

P

Parallel Library TCP/IP 87

pathmon 85

pathway.sh 85

performance

requirements 23

- small Project recommendations 46
- testing 52
- port numbers 46
- portal 18
- prefixes 40
- production, transition to 54
- professional services 24, 26
- project manager 27
- project plan 28

R

- RAM requirements
 - HP NonStop Server 62
 - HP Tru64 61
 - HP-UX 61
 - IBM AIX 61
 - IBM z/OS 62
 - Red Hat Linux 62
 - Sun Solaris 62
 - SuSE Linux 62
 - Windows 61
- Red Hat Linux
 - system requirements
 - CPU 62
 - disk space 62
 - RAM 62
- Repository
 - failover 64
- requirements 30
- requirements analysis
 - business planning needs 24
 - operation and performance needs 23
 - overview 21, 22
 - personnel and training needs 24
 - system-specific needs 22
- reusability 34
- risks 29

S

- scalability 45
- scheduling
 - deployment project plan 28, 29
 - test plan 31
- Screenshots 12
- security
 - requirements 23, 30
- SeeBeyond Consulting Services 24, 26
- service-oriented architecture
 - layers 35
 - overview 34
- signal 31 87
- SNMP Agent

- troubleshooting with 53
- SOAP 37
- speed testing 52
- stress testing 52
- subprojects
 - large development team 42
 - naming conventions 41
- suffixes 41
- Sun Solaris
 - system requirements
 - CPU 62
 - disk space 62
 - RAM 62
- SuSE Linux
 - system requirements
 - CPU 62
 - disk space 62
 - RAM 62
 - system requirements
 - determining 57
 - system testing 52

T

- technical requirements specification 30
- test plan 30, 50
- testing 49–53
 - acceptance testing 52
 - integration testing 51
 - performance testing 52
 - speed testing 52
 - stress testing 52
 - system testing 52
 - test plan 50
 - unit testing 51
- thread pool 46
- training 24
- Transaction Monitoring Facility 86
- transition to production 54
- troubleshooting
 - HP NonStop Server 87
 - pre-transition testing phase 53

U

- UDDI 37
- unit testing 51
- users
 - creating 39

V

- Version Control 42

Index

W

Windows

- CPU requirements **61**
- disk space requirements **61**
- RAM requirements **61**

Windows clustering

- Logical Host **73**
 - overview **63**
 - Repository **64**
 - work breakdown structure **28**
- WSDL **37**

X

- XML **37**