

# **Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide Release 2.0**

Document Number SC09-2798-00

December 14, 1998

Owners:  
IBM Corporation, <http://www.ibm.com>

Component Broker Home Page, <http://www.software.ibm.com/ad/cb/>



Component Broker for Windows NT and AIX

SC09-2798-00

**Planning, Performance, and Installation Guide**

Release 2.0





Component Broker for Windows NT and AIX

SC09-2798-00

**Planning, Performance, and Installation Guide**

Release 2.0

**Note**

Before using this information and the product it supports, be sure to read the general information under Appendix K, "Notices" on page 217.

**First Edition (December 1998)**

This edition applies to Release 2.0 of IBM Component Broker and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1997 1998. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>About This Book</b> .....	ix
Who Should Read This Book .....	ix
How This Book is Organized .....	ix
Documentation Conventions .....	xi
The Component Broker Documentation .....	xii

---

## Learning About Component Broker .....

1

<b>Chapter 1. Introduction</b> .....	3
The CBConnector .....	3
The Component Broker Server .....	3
Supported Clients .....	3
System Management .....	3
The System Manager .....	4
The System Management Agent .....	4
The System Manager User Interface .....	4
The CBToolkit .....	4
Samples .....	4
Object Builder .....	5
Object Level Trace and Remote Debugger .....	5

---

## Planning for Component Broker .....

7

<b>Chapter 2. Configuration and Capacity Planning</b> .....	9
Introduction .....	9
Dimensions of the Configuration .....	9
Application Information .....	10
Application Vital Statistics .....	10
Application Topology Constraints .....	10
Application Response Time Requirements .....	11
Application Workload Characteristics .....	11
Hardware Considerations .....	12
Assessing Memory Requirements .....	13
Assessing Disk Requirements .....	13
Assessing Single and Multi Processor Requirements .....	14
Operating System Topics .....	14
Windows NT .....	14
Calculating the Configuration .....	15
Typical Topologies .....	15
Simple Topology .....	16
Basic Workload Management Topology - Horizontal Topology .....	16
Vertical Topology .....	17
Advanced Topologies .....	18
Additional Options .....	18
Testing the Topology .....	18
An Estimation Technique .....	19
Creating the Installation Plan .....	20
Conclusion .....	21

<b>Chapter 3. Planning for the Component Broker Installation</b> . . . . .	23
Packaging . . . . .	23
System Requirements . . . . .	24
Prerequisites . . . . .	24
Component Broker Packages . . . . .	28
CBCConnector Compact Disc . . . . .	28
CBCConnector Clients Compact Disc . . . . .	29
CICS and IMS Application Adaptor Compact Disc . . . . .	29
Oracle Application Adaptor Compact Disc . . . . .	30
Supplemental Programs Compact Disc . . . . .	30
CBToolkit Compact Disc . . . . .	31
Installation Considerations . . . . .	32
Using the Prerequisite Checking Tool . . . . .	34
<hr/>	
<b>Installing Component Broker</b> . . . . .	37
<b>Chapter 4. Installing the Prerequisite Software on Windows NT</b> . . . . .	39
Verifying the Windows NT Version . . . . .	39
TCP/IP Considerations . . . . .	39
User ID Considerations . . . . .	40
For a DB2 Administrator ID . . . . .	40
For a Windows NT Administration ID . . . . .	40
Windows NT Configuration Considerations . . . . .	40
Installation Order . . . . .	41
Installing and Configuring DB2 . . . . .	41
Installing DB2 NT . . . . .	42
Installing the DB2 NT Server . . . . .	42
Configuring the DB2 Server for Remote Client Access . . . . .	43
Installing the DB2 Client . . . . .	44
Configuring the DB2 Client . . . . .	44
Configuring a DB2 TCP/IP Connection to an MVS Host . . . . .	45
Installing VisualAge C++ . . . . .	46
Installing VisualAge for Java . . . . .	47
Installing and Configuring DCE . . . . .	48
Installing the DCE 2.0 Server and Client . . . . .	48
Reinstalling DCE 2.0 (Both Server and Client) . . . . .	50
Installing the DCE Fix Pack . . . . .	51
Configuring DCE . . . . .	52
Configuring the DCE Server . . . . .	52
Configuring the DCE Client . . . . .	54
<b>Chapter 5. Installing the Prerequisite Software on AIX</b> . . . . .	57
Verifying the AIX Version . . . . .	57
AIX Configuration Considerations . . . . .	58
TCP/IP Considerations . . . . .	59
The User ID and \$HOME . . . . .	59
Installing and Configuring the Java Development Kit . . . . .	60
Installing and Configuring DB2 . . . . .	61
Installing the DB2 Server . . . . .	61
Step 1: Installing the Bundle Definitions . . . . .	62
Step 2: Installing the Software . . . . .	62
Installing the DB2 Software Developer's Kit . . . . .	63
Step 1: Installing the Bundle Definitions . . . . .	63



Step 2: Installing the Software	63
Configuring DB2 Server	64
Setting up the DB2 License	65
Configuring the DB2 Server for Remote Client Access	65
Installing the DB2 Client	66
Mounting and Creating the File Systems	66
Installing the DB2 Software Developer's Kit	67
Configuring the DB2 Client	67
Additional Configuration for Communication with the Server	68
Installing and Configuring DCE	69
Installing and Configuring the DCE Server	71
Installing the DCE Server	71
Configuring DCE Server	72
Installing and Configuring the DCE Client	72
Installing the DCE Client	73
Configuring the DCE Client	73
Installing C Set++	76
Installing the Fixes for the Prerequisite Products	76
Creating a CD-ROM File System	76
Installing the PTFs	77
<b>Chapter 6. Component Broker Migration</b>	<b>79</b>
Migrating Object Builder Models	79
Interface Changes That May Require Code Modifications	79
Release-to-Release Compatibility	80
Release-to-Release Interoperability	80
Migration Considerations for OS/390 Servers	80
<b>Chapter 7. Performing an Unattended Installation on Windows NT</b>	<b>81</b>
Preparing Your Response File	81
Running the Unattended Installation	83
Using the Log File	84
<b>Chapter 8. Installing Component Broker for Windows NT</b>	<b>85</b>
Installing Component Broker	85
Installation Options	86
Pre-Installation	86
Installing the System Manager Option	87
Configuring the System Manager Option	89
Installing the Standalone Workstation	92
Configuring the Standalone Workstation	93
Installing the Application Server Option	96
Configuring the Application Server	98
Installing the Typical Clients Option	101
Configuring the Typical Clients	102
Installing the Custom Install Option	103
Configuring the Custom Install	105
Installing the Development Environment	108
Applying the Component Broker PTF	110
Stop the Component Broker Servers	111
Apply the Component Broker PTF	111
Restart the System Management Service	111
Accessing the Online Documentation	112
Adding Components to an Existing Installation	112

Reconfiguring Your Component Broker Installation	113
Uninstalling Component Broker	117
Uninstall Failure Cleanup	117
<b>Chapter 9. Installing Component Broker for AIX</b>	<b>121</b>
Installing the Server Run Time	121
Installing the Client Run Time	123
Configuring Your Component Broker User ID	125
Configuring the Run Time	126
Reconfiguring the Run Time	127
Installing the Development Environment	127
Applying the Component Broker PTF	128
Stop the System Management Service	128
Apply the Component Broker PTF	129
Restart the System Management Service	130
Accessing the Online Documentation	130
Uninstalling Component Broker	130
<b>Chapter 10. Installing the Component Broker Client for Windows 95</b>	<b>133</b>
Installing the Java Client	133
Uninstalling the Java Client	134
Recovering from an Uninstall Failure	134
<b>Chapter 11. Installing the Component Broker Client for OS/2</b>	<b>135</b>
Installing the Java Client	135
Uninstalling the Java Client	135
<b>Chapter 12. Installing the Component Broker Java Client on a Web Server</b>	<b>137</b>
Using a Downloaded Java Client	138
<b>Chapter 13. Problem Determination</b>	<b>139</b>
Understanding Install or Uninstall Failures	139
Setup Unable to Find Domain Name	139
Coexisting With the Hummingbird SOCK5 Server	140
Manually Clearing Up the Existing Configuration	140
Unable to Start the Component Broker Configuration Tool	141
Return Code 1069 From System Manager Configuration	142
Incomplete PATH After Rebooting Component Broker or Installing Another Product	142
Database CBSRFCDB Not Created Correctly	143
<b>Chapter 14. Testing the Component Broker Installation</b>	<b>145</b>
The Initial Configuration	145
The Policy Sample Application	145
Configuring the Policy Sample Application	146
Define the PolicyFamily Application to System Management	146
Configure an Application Server	147
Load the Application on a Server	147
Executing the Policy Sample Application	148
Activating Your Configuration	148
Running the Sample Application	148
Verifying Services With DCE	149
Reloading an Application	150
Adding Objects to a Server	150
Disabling Security Service	151

<b>Chapter 15. Information About Component Broker Samples</b>	153
General Information	153
Installation Verification Programs	153
Tutorial Samples	153
Early Test Function	154
Documentation for the Samples	154
Generating the Business Objects for the Samples	155
Compiling and Running the Samples	156
Programming Model and JDK Code Examples	156
<hr/>	
<b>Performance and Tuning</b>	157
<hr/>	
<b>Chapter 16. Performance Monitoring and Tuning</b>	159
CBConnector Tuning	159
Operating System Tuning	160
Java Virtual Machine Tuning	161
Database Tuning	163
Performance Monitoring	163
<hr/>	
<b>Appendixes</b>	167
<hr/>	
<b>Appendix A. Sample Response File</b>	169
<hr/>	
<b>Appendix B. Advanced DB2 Concerns</b>	173
Dropping an Existing Database on a DB2 Server	173
Installing a Component Broker Server on a DB2 Client	173
Advanced Access to DB2	175
Configuring DB2 For Use with Component Broker	176
Next Key	176
Settings of Timeout Values	176
Heap Size	177
DB2 Transaction Logs	178
<hr/>	
<b>Appendix C. Installing a Sample Configuration Using UDB</b>	181
Install and Configure Host1	181
Install Host2	182
Install and Configure Host3	182
Install and Configure Host4	184
Install Host5	186
Summary	187
<hr/>	
<b>Appendix D. The Two-Compiler Environment (NT only)</b>	189
Creating Header Directory for ActiveX Client	189
Preparing Separate Environments	189
The VisualAge C++ Environment	190
The Microsoft Visual C++ Environment	190
<hr/>	
<b>Appendix E. Enabling the Security Service</b>	191
Enable DCE Security for Servers in a Server Group	191
Enable DCE Security for the C++ Client in a Management Zone	192
Activate DCE Security Enablement for Servers and Clients	192
Enable SSL Security for Servers in a Server Group	192

Activate SSL Security Enablement for Servers in a Server Group . . . . .	193
Enable Java Client With DCE Security . . . . .	194
Enable Java Client With SSL Security . . . . .	195
Create Your Own Client Properties File . . . . .	195
Enable Application or Applet with SSL Security . . . . .	196
Troubleshooting Security Problems . . . . .	197
<b>Appendix F. File Locations and the Java Classpath . . . . .</b>	<b>199</b>
Additional Java Considerations . . . . .	201
<b>Appendix G. Java and Just in Time (JIT) in Component Broker . . . . .</b>	<b>203</b>
Using JIT on Component Broker for Windows NT . . . . .	203
Using JIT on Component Broker for AIX . . . . .	204
Using JIT on Component Broker for OS/2 and Windows 95 . . . . .	205
<b>Appendix H. Environment Variables . . . . .</b>	<b>209</b>
Windows NT Platform . . . . .	209
AIX Platform . . . . .	209
Environment Variables . . . . .	209
<b>Appendix I. Starting and Restarting Servers . . . . .</b>	<b>213</b>
<b>Appendix J. Stopping Servers . . . . .</b>	<b>215</b>
<b>Appendix K. Notices . . . . .</b>	<b>217</b>
Trademarks . . . . .	218

---

## About This Book

This book provides information on how to plan and configure the Component Broker environment to be used for a variety of purposes. This includes single standalone development and run-time machines, development environments that have a mixture of software development toolkits and run times, and run-time production configurations where the products of development (Component Broker Applications) can be managed and executed.

Do not use the *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide* as a substitute for the Component Broker library. The library provides detailed information beyond the concepts introduced in this book. Before installing Component Broker, read the README.HTM file, located in the Component Broker root directory, and the Late Breaking News memo in printed form for last minute product information. After completing the tasks and exercises in this book, review the rest of the product library.

---

## Who Should Read This Book

The *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide* is intended primarily for administrators and coordinators of advanced development environments who want to:

- Plan and install an environment intended for multi-user development
- Plan and install an environment intended for production application deployment
- Order hardware and prerequisite software for either of the previously mentioned environments
- Understand configuration options for prerequisite software
- Evolve and tune Component Broker deployment environments

First time users of Component Broker or developers wishing to work in a single machine development environment should refer to the *Component Broker for Windows NT and AIX Quick Beginnings* and use this book only as a reference.

---

## How This Book is Organized

Chapter 1, "Introduction" on page 3 provides a introduction to the functions and features of Component Broker as they apply to this document.

Chapter 2, "Configuration and Capacity Planning" on page 9 provides deployment configuration guidelines, specifications and formulas. This includes information about memory and CPU requirements for running various Component Broker run-time environments.

Chapter 3, "Planning for the Component Broker Installation" on page 23 describes the required system environment and software prerequisites for installing the CBConnector package and the CBToolkit package. This chapter describes which installable units are part of each Component Broker package.

WIN Chapter 4, "Installing the Prerequisite Software on Windows NT" on page 39 provides information about installing and configuring the prerequisite products on a computer running Windows NT.

AIX Chapter 5, "Installing the Prerequisite Software on AIX" on page 57 provides information about installing and configuring the prerequisite products on a computer running AIX.

Chapter 6, “Component Broker Migration” on page 79 details how and when to migrate from a previous version of Component Broker.

Chapter 7, “Performing an Unattended Installation on Windows NT” on page 81 provides information about running an unattended installation of the Component Broker packages on a Windows NT computer.

**WIN** Chapter 8, “Installing Component Broker for Windows NT” on page 85 provides the procedures for installing, configuring, and uninstalling Component Broker for Windows NT. This chapter includes Information about customizing the default response file.

**AIX** Chapter 9, “Installing Component Broker for AIX” on page 121 provides the procedures for installing, configuring, and uninstalling Component Broker for AIX and its prerequisite software.

**WIN** Chapter 10, “Installing the Component Broker Client for Windows 95” on page 133 provides the procedures for installing and uninstalling the Component Broker client for Windows 95.

**OS/2** Chapter 11, “Installing the Component Broker Client for OS/2” on page 135 provides the procedures for installing and uninstalling the Component Broker client for OS/2.

Chapter 12, “Installing the Component Broker Java Client on a Web Server” on page 137 provides the procedure to install the Component Broker Java client on a Web server as a downloaded Java applet.

Chapter 13, “Problem Determination” on page 139 describes recovery procedures for problems that might occur during Component Broker installation.

Chapter 14, “Testing the Component Broker Installation” on page 145 describes common processes that use Component Broker system management. These processes include starting and stopping servers, executing samples, and verifying the installation. For detailed information on system management, see the *Component Broker for Windows NT and AIX System Administration Guide*.

Chapter 15, “Information About Component Broker Samples” on page 153 provides information about the samples provided with Component Broker.

Chapter 16, “Performance Monitoring and Tuning” on page 159 provides information about Component Broker performance, and provides tips to guide you in tuning your installation.

Appendix A, “Sample Response File” on page 169 provides an example of a typical response file.

Appendix B, “Advanced DB2 Concerns” on page 173 provides information on installing a Component Broker server on a DB2 client system and information on advanced access to DB2.

Appendix C, “Installing a Sample Configuration Using UDB” on page 181 discusses procedures to create a five-computer configuration using UDB.

**WIN** Appendix D, “The Two-Compiler Environment (NT only)” on page 189 provides information and procedures for creating a two-compiler environment. A two-compiler environment is required for any Component Broker system that installs the ActiveX client with either the C++ or Java client.

Appendix E, “Enabling the Security Service” on page 191 discusses enabling clients, servers, and applications with DCE or SSL security. This appendix includes a section for troubleshooting security problems.

Appendix F, “File Locations and the Java Classpath” on page 199 discusses the Java classpath and the environments under which specific files should be listed in the classpath.

Appendix G, “Java and Just in Time (JIT) in Component Broker” on page 203 discusses the interactions between Java and Just in Time on Windows NT, AIX, Windows 95, and OS/2.

Appendix H, “Environment Variables” on page 209 provides a list of environment variables used by Component Broker.

Appendix I, “Starting and Restarting Servers” on page 213 provides a procedure you can use to manually start or restart one or more servers.

Appendix J, “Stopping Servers” on page 215 provides a procedure for stopping a server.

---

## Documentation Conventions

The following conventions distinguish different text elements.

**plain** Window titles, folder names, icon names, and method names.

**monospace** Programming examples, user input at the command line prompt or into an entry field, directory paths, and user output.

**bold** Menu choices and menu names, labels for push buttons, check boxes, radio buttons, group-box controls, drop-down list boxes, combo-boxes, notebook tabs, and entry fields.

*italics* Programming keywords and variables, titles of documents, and initial use of terms that are in the glossary.

The following short cut conventions are used to abbreviate menu selections and object expansions within tree views of the System Manager User Interface and Object Builder.

→ The right arrow when used within a menu shows a series of menu selections. For example, “**File** → **New**” is translated to mean: “On the **File** menu, click **New**”.

The right arrow when used within a tree view shows a series of folder (or object) expansions. For example, “Expand Management Zones → Sample Cell and Work Group Zone → Configuration” is translated to mean:

1. Expand Management Zones.
2. Expand Sample Cell and Work Group Zone.
3. Expand Configuration.

**Note:** An object in a view can be expanded when there is a plus sign (+) beside that object. After an object is expanded, the plus sign is replaced by a minus sign (-).

**+** Expands a tree structure to show more objects. To expand, click the plus sign (+) beside any object. If you double-click the object, a new tree structure is displayed with that object as the root of the tree.

**-** Collapses a tree structure to review from view its containing objects. To collapse, click the minus sign (-) beside any object.

### left mouse button

Used for all actions in the application except for opening the pop-up menu of an object. For example, if you click with the left mouse button on an object, it is selected. Click with the left mouse button on a menu option to perform that action.

### right mouse button

Opens the pop-up menu of an object that contains a list of actions that can be performed on that object. The list varies depending on the type of object.

WIN

Denotes a section that applies only to the Windows 95 or Windows NT platform. Do not interpret this symbol to denote that an equivalent AIX section exists.

**Note:** The Windows 95 platform only supports the Component Broker Java client.

AIX

Denotes a section that applies only to the AIX platform. Do not interpret this symbol to denote that an equivalent Windows section exists.

OS/2

Denotes a section that applies only to the OS/2 platform.

**Note:** The OS/2 platform only supports the Component Broker Java client.

---

## The Component Broker Documentation

The following information is part of Component Broker for Windows NT and AIX:

- Help information is available from Component Broker product panels.
- The Component Broker online library can be viewed using a frames-compatible Web browser:  
[http://localhost:49213/cgi-bin/cbwebx.exe/en\\_US/cbdoc/Extract/0/index.htm](http://localhost:49213/cgi-bin/cbwebx.exe/en_US/cbdoc/Extract/0/index.htm)
- *Component Broker for Windows NT and AIX Quick Beginnings*, G04L-2375 explains how to easily create and verify a starter Component Broker environment. These instructions walk the user through a typical server and client installation. Users can extend this configuration using the information in the *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide*.
- *Component Broker for Windows NT and AIX CICS and IMS Application Adaptor Quick Beginnings*, GC09-2703 provides a brief technical overview of the CICS and IMS application adaptor and guides the user through its installation and configuration. Step-by-step instructions guide the user through creating an initial CICS and IMS application using application development tools included in the CBToolkit package.
- *Component Broker for Windows NT and AIX Oracle Application Adaptor*, GC09-2733 provides a brief technical overview of the Oracle application adaptor and guides the user through its installation and configuration. Step-by-step instructions guide the user through creating an initial Oracle application using application development tools included in the CBToolkit package.
- *Component Broker for Windows NT and AIX System Administration Guide*, SC09-2704 provides information about configuring and operating one or more hosts managed by Component Broker. It also provides general information about using the System Manager User Interface.
- *Component Broker Application Development Tools Guide*, SC09-2705 explains how to create and test Component Broker applications using the tools provided in the CBToolkit with a focus on common development scenarios such as inheritance and team development.
- *Component Broker Programming Guide*, G04L-2376 describes the programming model including business objects, data objects, and information about MOFW, IDL, and C++ CORBA programming.
- *Component Broker Advanced Programming Guide*, SC09-2708 describes the Component Broker implementation for the CORBA Object Services and the Component Broker Object Request Broker (including remote method invocation and the Dynamic Invocation Interface (DII) procedures), Session Service, Cache Service, Notification Service, Interlanguage Object Model (IOM), and work-load management (WLM).
- *Component Broker Programming Reference*, SC09-2810 contains information about the APIs available to Component Broker application developers.



- *Component Broker for Windows NT and AIX Problem Determination Guide*, SC09-2799-00 explains how to identify and resolve problems within a Component Broker environment using the tools provided with Component Broker. The book includes information on installation problems, run time errors, debugging of applications, and analysis of log messages.
- *Component Broker Glossary*, SC09-2710 contains terms and definitions relating to Component Broker.
- *OS/390 Component Broker Introduction*, GA22-7324 describes the concepts and facilities of Component Broker and the value it has on the OS/390 platform. The audience is a knowledgeable decision maker or a system programmer.
- *OS/390 Component Broker Planning and Installation*, GA22-7331 describes the planning and installation considerations for Component Broker on OS/390.
- *OS/390 Component Broker System Administration*, GA22-7328 describes system administration tasks and operations tasks, as provided in the system administration user interface for OS/390.
- *OS/390 Component Broker Programming: Assembling Applications*, GA22-7326 provides information for assembling applications using Component Broker on OS/390.
- *OS/390 Component Broker Operations: Messages and Diagnosis*, GA22-7329 provides diagnosis information and describes the messages associated with Component Broker on OS/390.



---

# Learning About Component Broker



---

## Chapter 1. Introduction

This chapter provides an introduction to the major features needed to plan, install, configure, and test a Component Broker installation.

Component Broker is a middleware product that can be used to design, develop, and deploy distributed object-oriented applications. Deployed applications rely on middleware services and interfaces to implement business processes.

Component Broker provides a server run time where object-oriented server applications can be run. Objects can be developed and later installed on the Component Broker servers. These servers support other server applications and non-Component Broker client applications that use the installed classes and objects. Component Broker has integrated system management that supports definition, configuration, installation, operation, and monitoring of Component Broker servers and object-oriented applications installed on servers.

The CBToolkit components provide application development tools that simplify the development process for distributed object-oriented applications.

---

### The CBConnector

The primary Component Broker Connector (CBConnector) installable units shipped on the CBConnector compact disc are:


- The Component Broker Server
- Supported Clients
- System Management

### The Component Broker Server

The Component Broker server manages the objects used by clients running Component Broker applications. Component Broker objects run in conjunction with the Component Broker server in an operating system process like any compiled code.

### Supported Clients

Component Broker provides the following clients.

- C++
- Java
-  ActiveX

### System Management

Component Broker system management consists of the following parts.

- The System Manager
- The System Management Agent
- The System Manager User Interface

An administrator uses the System Manager User Interface to manage the clients and servers within a management zone. A management zone consists of the following:

- Only one System Manager.
- Zero or more System Management Agents.
- One or more System Manager User Interfaces.
- One or more Component Broker servers.
- Zero or more supported clients.

Within a management zone, a single server may not be able to handle the implementation of a business application. If this occurs, multiple servers could be required to handle the availability, scalability, and performance requirements.

## The System Manager

The System Manager controls the servers and clients within a management zone. The System Manager provides the logic required to manage a configuration of Component Broker servers, clients, associated objects, and applications.

## The System Management Agent

There is one System Management Agent installed on each centrally managed host. The System Management Agent is the way the System Manager communicates with the objects to be managed on that host. The System Management Agent interacts directly with those objects, passing data back to the System Manager and acting on the objects on behalf of the System Manager.

**Note:** A System Management Agent cannot be installed on the host containing the System Manager for the management zone.

## The System Manager User Interface

The System Manager User Interface can be installed on any host and does not need to co-reside with a System Manager or a System Management Agent. There can be multiple System Manager User Interfaces installed and running in an enterprise. However, each System Manager User Interface must be connected to only one System Manager.

---

## The CBToolkit

The Component Broker Toolkit (CBToolkit) is a collection of visual application development tools that help to define, implement, deploy, and debug distributed Component Broker applications.

The primary tools of the CBToolkit are:

- Samples
- Object Builder
- Object Level Trace and Remote Debugger

## Samples

Several sample applications are provided to help demonstrate the features in the CBConnector. These samples consist of installation verification programs, tutorial samples, and early test function. The installation verification programs and the tutorial samples demonstrate the Object Builder tool. For more information, refer to Chapter 15, “Information About Component Broker Samples” on page 153.

## Object Builder

Object Builder is a visual programming and application development tool. The data entered in the provided smart guides is used by the code generation templates to emit most of the implementation of a component. With Object Builder, only user-defined methods need to be defined for implementation.

Application development in Object Builder consists of defining, building, and packaging components. These components can be defined in any of the following ways:

### **From a new design**

The design of these components start from the component interface and works down to the component datastore. This type of design is called “top-down.”

### **From an existing database**

The design of these components start from the datastore and works up to the component interface. This type of design is called “bottom-up.”

### **From a combination of the previous design strategies**

The design of these components is a combination of creating a new component and data interface with an existing datastore. This type of design is called “meet-in-the-middle.”

Object Builder generates the necessary code for the objects and relationships that are defined, including framework methods and database calls. Additionally, Object Builder handles the building and packaging details for the applications that are created.

## Object Level Trace and Remote Debugger

The toolkit provides both a local debugger and Object Level Trace (OLT), a remote debugging tool for testing deployed applications.

OLT monitors client-server communication and provides a graphical trace display from which breakpoints can be set.

The OLT Remote Debugger debugs server and client applications seamlessly from one machine, as though they were a single application.





---

# Planning for Component Broker



---

## Chapter 2. Configuration and Capacity Planning

This section includes the following topics:

- “Introduction”
- “Dimensions of the Configuration”
- “Calculating the Configuration” on page 15
- “An Estimation Technique” on page 19
- “Creating the Installation Plan” on page 20
- “Conclusion” on page 21

---

### Introduction

Before beginning to install a Component Broker production deployment environment or a team development environment, you should first perform some careful planning. This chapter provides some heuristics dealing with capacity and configuration planning.

This chapter is intended to help system administrators plan and prepare for a Component Broker deployment environment by determining computer sizes and configurations. To accomplish this, you must address a number of questions about the production deployment. This chapter cannot provide all the answers. Some answers must come from the environment into which the Component Broker application is to be placed. Other answers begin to emerge as specifics about the application and usage patterns are acquired.

A secondary goal of this chapter is to provide some guidance on how to plan for and set up integration test and preproduction environments. Most of the same heuristics can be applied. Additional details unique to a development environment are provided when appropriate.

Another goal of this chapter is to provide information about how Component Broker works in specific areas that impact and effect capacity and throughput as well as configuration choices. Therefore this chapter could be valuable to developers who are involved in other aspects of building and deploying Component Broker applications.

---

### Dimensions of the Configuration

If you install the System Manager option of the latest release of Component Broker, what you will have is anything but typical for most situations. To date, the focus in Component Broker has been to enable and optimize the documentation and installation tools around the “typical developer” view of things. While individual developers will continue to use the Standalone workstation installation as the basis for stand alone development work in Component Broker, more and more attention is being paid to alternate configurations. A deployment configuration, for purposes of this chapter, is defined as any combination of Component Broker run times installed on a set of coordinated and managed hosts that are necessary to enable the execution of a real Component Broker application. The word “coordinated” is very important because it suggests a set of interrelationships between hosts. It begins to give clues about DCE configuration, third tier adapter configurations, systems management zones and systems management configurations.

Setting up and configuring the deployment configuration requires you to make many decisions about which computers are used for which purpose and which parts of the application are placed on the various computers. This section first describes “application information” which you must understand. That is followed by a discussion of the hardware configurations and operating systems involved in the deployment

configuration. This section gathers the inputs or raw material from which the deployment configuration can be described and created.

## Application Information

To further define and understand the nature of the deployment configuration, you must collect additional data about the application. The following sections describe what you need in terms of vital application statistics. These include an assessment and enumeration of topology constraints, expected workload and throughput as well as the nature of the workload implied by the application's use cases.

### Application Vital Statistics

Proper and accurate capacity and configuration planning can only be done if the application to be deployed is understood in great detail. You must answer the following questions:

1. How many managed objects are there?
2. What kinds of business objects are in the application?
  - How many application objects are there?
  - How many composed business objects are there?
  - How many basic business objects are there?
3. How many applications are in the application family?
4. How many executable sets are there?
5. What kind of clients are being used?
  - Java (applet, application, servlet)?
  - C++?
  - ActiveX?

### Application Topology Constraints

After you gather vital statistics, you must understand the application topology constraints. It is very important to clearly understand the constraints that exist in the application to be deployed. Constraints generally arise because of the particular application adapters which are being leveraged. It usually is not the application adapter code itself that drives the constraint. Instead, it is often the nature of the distribution aspects that are associated with the underlying resource managers being used.

The most prevalent constraints in the application topology are those imposed by the third tier resource manager. In the relational database case, this set of constraints revolves around the placement of tables into databases. This involves the following questions:

1. How many databases are used in the application being configured?
2. How many tables are in each database?
3. What are the relationships between objects in the object model?
4. How many of them are manifested using foreign keys in the data objects?

Foreign key mappings are a common and efficient way to resolve one-to-one and one-to-N relations. This pattern does suggest a set of guidelines relating to the application topology. The relationships between mapping patterns and topology are described next, starting with the optimal topologies and patterns, then briefly explaining what is possible and what the performance implications are of the non-optimal configurations.

To achieve best performance, the database design should be such that related tables are in the same database and use a foreign key pattern to implement the object relationship. This approach allows

navigation between related objects or queries that navigate or join related objects without requiring extra database connections and associated commit processing. The query service can have the database manager perform the join processing between related tables and is able to take advantage of database indexes for better performance.

If this is not possible, perhaps because of legacy database considerations or capacity constraints of a single database or database server, then you can still use the foreign key pattern to implement the object relationship across tables in different databases. You do not lose the ability to navigate between related object or to perform queries. However table joins will have to be performed by the query service rather than by the database manager resulting in some performance loss. Extra connections and commit processing are also required. From a topology perspective, creating new multidatabase relationships is not desirable. From design and integration of legacy perspective, it might be necessary.

Implementing object relationships using handles or SORs should be used when the foreign key pattern cannot be used. This may involve relationships between objects where the object being pointed to is not backed by a relational database.

Based on these guidelines, decisions about distribution of objects onto servers and the physical computer topology used can be made. Additional information on the performance considerations associated with relational databases is found in Chapter 16, “Performance Monitoring and Tuning” on page 159.

These are the most direct topology constraints and the first you must understand. Beyond that it is important to understand the logon scheme that is used. You need to ask if the same user ID and password are used for accessing all the tables in the database or all the databases involved in the application? The answers have some downstream effects when it comes to containers leveraged, distribution of function to servers on a host, and usage of multiple hosts in the configuration.

Is it easy to provide access to the third tier resource manager for multiple Component Broker hosts that might have servers on them? If the answer is no, then the number of hosts used in the topology must be fixed or at least considered a more permanent part of the configuration.

Some third tier resource managers may impose restrictions on the number and types of connections which are allowed for a given client host (in this case the Component Broker Server Host). Information like this can constrain the topology to one application server per host. It is important to know this before buying, for example, a four-way multiprocessor computer designed to run many application servers.

## Application Response Time Requirements

If there are specific response time requirements for uses cases, either as a group or as a whole, you need to write them down and factor them into the capacity and configuration planning. Response time is one of the things upon which the effectiveness of the Component Broker application will be judged by the end users. End users usually do not need to know about all of the objects, but they want the application to work reliably within response time requirements.

## Application Workload Characteristics

The next set of data you must gather deals with workload characteristics.

If there is a **number of use-cases** or application objects associated with the design, then there is already a starting point from which to determine the workload. The first rough approximation of the workload is the **average number of active users** and the **mix of uses cases that are being executed** at any given time. This information can all be gathered without doing any hands-on experimentation. It is just a model of the workload.

In most business situations, the **number of active users** and the **mix of uses cases** that are being executed can vary over the course of a 24-hour period. Therefore, it is more desirable to have details on the workload on an hour-by-hour basis. It is often the case that use cases like “login” or “sign-on” occur as a spike early in a normal business day. As you will see later, this may or may not be a problem depending on the amount of work that goes on in the uses cases which are part of the period where the system is spiking.

The above information is the bare minimum required to create a capacity and configuration plan. More data, however, increases the reliability of the estimates. The following paragraphs describe additional workload information that is highly valuable when doing capacity planning.

What are the active users doing in the use-case or more simply, where is their time spent? A simple model of **time spent per use case** (in isolation or on a server under load) is a desirable fact to obtain. From there, one should determine how much time is spent in the Component Broker servers versus how much is spent running to third tier resource managers. A Component Broker server running use cases which spend seventy-five percent of the time waiting for a response from a mainframe transaction can handle more load and should be configured differently than a server running use cases which spend seventy-five percent of the time in the object server doing business logic that is contained in the business objects.

Now that you know and have cataloged the available details of the application to be deployed, it is time to consider the hardware and base operating system. The following sections detail relevant hardware and operating system topics related to capacity planning. It also sets some baselines for Component Broker hosts which are to participate in the deployment configuration.

## Hardware Considerations

There's always two sorts of questions when it comes to hardware and deployment configurations. They are:

1. **I need to order some computers. Which ones should I order?** This is the lucky case and the most fun for administrators and operators.
2. **I have the following set of computers. Are they useful when running a Component Broker deployment environment?** This is the more complicated case. The answer is most probably yes. Component Broker is a distributed object base system and can usually find an effective role for most reasonably sized existing systems.

These questions will be addressed in order. First, however, some basic guidelines and assumptions about hardware utilization are enumerated and justified.

When building a deployment configuration from scratch, the following recommendations about topology apply:

- **Plan on having the DCE servers on a separate host.** For planning purposes, assume that the DCE Servers are all on one separate host. That means the Cell Directory Service (CDS), the Security Server, and other DCE Servers all reside on one DCE Server computer. Remember that the deployment configuration for a given application or set of applications must run under one DCE cell.
- **Prefer physical separation of third tier.** Given a choice, it is usually better to have the DB2 Server, for example, on a separate computer from hosts where Component Broker Servers are to be run. This simplifies management and enables more effective scaling when that need arises. When using MVS or other host services that require operating systems that Component Broker does not support, this choice is made already.

- **Prefer a separate computer to host the cell and work group name tree and systems management duties.** It is not preferable to have the same computer playing the role of System Manager and application server.

What kinds of computers are optimal to play any of the roles (for example, DB2 Server, DCE Server or Component Broker Server) implied above? Computers that can best play these roles should have:

1. Adequate physical memory to minimize high page faulting rates.
2. Disks that are SCSI attached to minimize CPU usage along with access times that are less than ten milliseconds.
3. Processor speeds sufficient to handle incoming workload rates.

## Assessing Memory Requirements

A memory simulator allowing you to test a range of memory sizes while your application is running is the best method for determining the optimum memory size for your system. This methodology allows you to determine how many megabytes of real memory your system needs to run Component Broker to achieve an acceptable level of performance.

On Windows NT, an available memory simulator may be found in the NT Resource Kit. The AIX Performance Tools supplies the `rmss` command. Details on running each of these tools can be found in the corresponding documents (the *Microsoft Windows NT Workstation Resource Kit* and the *AIX Performance Tuning Guide Versions 3.2 and 4*).

The simulation process runs your application under a typical workload and adjusts the memory size until the page faulting rate is minimized. Ideally you would start out with excess memory and use the simulation tools to work your way down to a reasonable configuration.

If you do not have access to these tools, the following guidelines will help until you are able to do the simulation runs.

$$\begin{aligned} \text{Memory requirements} = & \text{Base memory configuration} + \\ & 40 \text{ Mb} * \text{Number of Application Servers} + \\ & 40 \text{ Mb} * 1 \text{ Name Server} + \\ & \text{Total size of Application DLL's} + \\ & \text{3rd party software memory requirements (for example, DCE, DB2)} \end{aligned}$$

Where:

- Windows NT Base memory configuration is equal to 128 megabytes
- AIX Base memory configuration is equal to 256 megabytes

## Assessing Disk Requirements

Component Broker servers use the disk for a number of things, the big items being transaction logging and Component Broker configuration data storage. The logging is necessary to provide reliable transaction services in the event of system failures. While the data throughput needs of the disk are not extensive, fast access times and minimizing CPU usage are important items to consider.

SCSI disk and adapter configurations offer superior performance to help achieve these goals.

Access times of less than ten milliseconds and twenty megabyte throughput rates are not unreasonable for a starter configuration.

From a disk footprint perspective, Component Broker needs approximately 200 megabytes of disk space to hold the software that runs the Component Broker server. This is a minimal figure, and you should

consider setting a virtual memory paging size that is around three times the size of the real memory on the computer. This can take up a sizable amount of disk space if you are running computers in excess of 256 megabytes of main memory. If you are running DCE and DB2 on the same computer, the disk space requirements are higher.

Component Broker also uses the disk for transaction logs, error logs, and activity logs. These are usually all quite small even in a production configuration. In general, for a typical server that is playing no third tier roles, four gigabytes of disk space is all that is required. There's always an advantage to having multiple physical disks. Distributing the I/O intensive operations across disks is a good thing to do. For example, having the virtual memory paging space on one disk and the Component Broker transaction logs on another is preferred over having them all on one physical disk.

## Assessing Single and Multi Processor Requirements

Component Broker can use a fast processor, especially when many users are active and doing work at the same time. Component Broker will probably swamp or overload the processor before memory if your physical memory exceeds 256 megabytes unless there are 200 or more active users on the server that are waiting on third tier resource managers.

The above discussion of processors has focused on single-processor systems. These are sufficient and adequate, but Component Broker is also able to leverage multiprocessor systems. Nothing special has been done to Component Broker to run with multiprocessor computers. However, on multiprocessor computers, it is recommended that at least 2 application servers be leveraged if the application topology considerations allow it. Multiple servers will find their way to a processor each. See Chapter 16, "Performance Monitoring and Tuning" on page 159 for information about how to optimally configure your servers on a multiprocessor system.

What does two servers per host really mean? The Component Broker Name Server also runs as a Component Broker server. The Name Server does not run any user business objects, but it does run MOFW objects that support system functions. If you have optimized your usage of DCE (or the Name Server) then it does not need to count as an application server. If you have not optimized DCE usage, then you need to count the Name Server as another server when allocating servers to processors.

After determining the hardware requirements, the next decisions you must make concern the operating system type and configuration. The next section addresses these.

## Operating System Topics

This section includes the following topics:

- "Windows NT"

### Windows NT

Component Broker currently supports Windows NT 4.0 Workstation and Server with a mandatory prerequisite of SP3 (service pack 3). This section discusses whether Component Broker uses the Windows NT Workstation or the Windows NT Server.

Each of the Windows NT operating system packages can be successfully used to deploy Component Broker applications. There are some compelling reasons why the Windows NT Server is the right choice in most situations. The Windows NT Server 4.0 provides:

- Lots of functions that are not required to run Component Broker.
- Optimizations for server workloads in the following areas:



- A write throttling mechanism (the point at which dirty data is flushed to disk) is implemented differently for the Windows NT Server and the Windows NT Workstation. This allows the server to handle server-like loads.
- The internal code that provides the Windows NT server functionality (SRV.SYS) is less pageable. This leads to better response times under heavy server-type loads.
- Preloads. Under the Windows NT workstation, the Windows NT Virtual DOS Computer is preloaded for improved application start-up times. The Windows NT Server does not preload the VDM, but instead optimizes server file operations.
- Threading. The Windows NT Workstation and the Windows NT Server use a different number of system worker threads for blocking operations. The Windows NT Server is optimized for fast server type I/O operations while the Windows NT Workstation is optimized for desktop applications.
- Flexibility. The Windows NT Server can be optimized for file and print server performance or application server performance, allowing you to tune it for maximum throughput depending on the type of server usage. The Windows NT Workstation performance is about the same as the Windows NT Server up to approximately 10 different simultaneous inbound user connections. This enables the Windows NT Workstation to function well in a small peer environment. However, the Windows NT Server is the correct choice if you want to deploy a server in a larger environment or want to scale your network in the future. Additional details are available on the Microsoft web pages.

---

## Calculating the Configuration

By gathering the data that was discussed in the previous sections, you now have the basic information to calculate the proper configuration topology. This section is based on the data gathered in response to questions in previous sections and some estimation techniques. Results generated by following the techniques described in this section should be considered a starting point. There is no substitute for additional testing of a system under load. This section should, however, help you establish a good initial configuration from which to complete more sophisticated load tests that can be used to refine and tune the initial configuration.

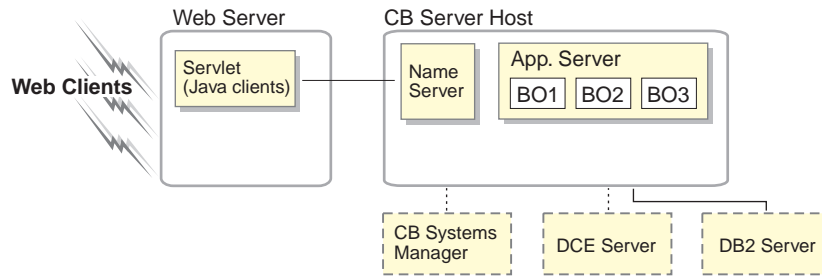
A complete performance model of Component Broker that could serve as a basis from which you could construct a performance model of a Component Broker application is not available at this time.

## Typical Topologies

A model is a very complete and scalable formula from which the proper configurations can be calculated. One of the first steps toward establishing a model is to focus on taking some measurements on key topologies for what are expected to be prototypical workloads. There are a number of prototypical topologies in which Component Broker applications will be deployed. These are described in the following sections.

## Simple Topology

The first and simplest prototypical topology runs all business objects on the same Component Broker host



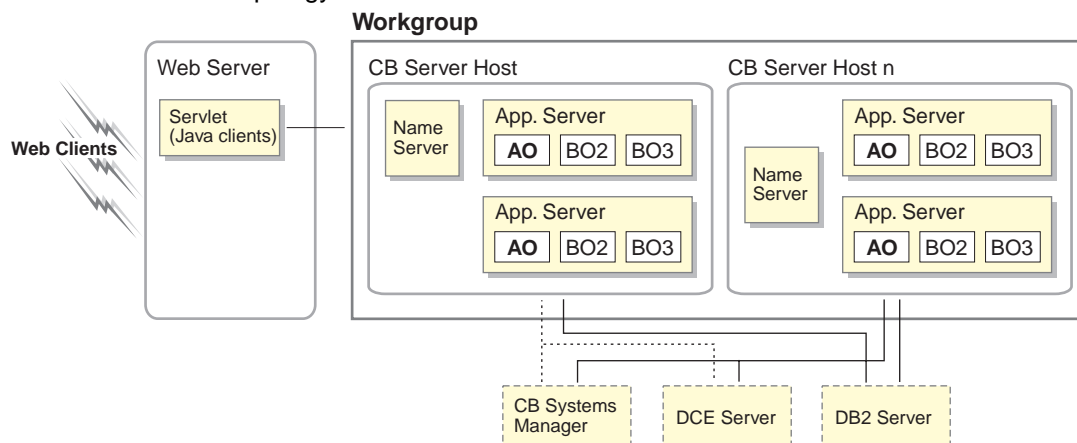
using a single server.

Even this simple topology shows a number of hosts involved. By following the recommendations described previously, there are three supporting hosts. The Component Broker Systems Manager runs a Component Broker installation and manages the Component Broker server host that has the application server. The DCE server and the DB2 server do not run any Component Broker software, but are basic to a deployment configuration. The DB2 server in this picture represents a heavily used third tier resource manager. Specific application topologies will vary and this could be a System/390 running DB2, IMS, or CICS as well as any other platform that is playing the role of logical and physical third tier.

This simple topology represents a baseline. There is no separation of business logic across computers. There are certainly a minimum number of calls between hosts and processes. Scaling this topology can go in several different directions. The most obvious and easiest scaling tactic is to make the single host bigger and more powerful. However, this does not take advantage of many of the features of Component Broker or of the inherent flexibility of distributed objects. The following topologies address this topic.

## Basic Workload Management Topology - Horizontal Topology

The second topology introduces workload management as an addition to the first topology. All business objects run on a single server, but in fact there are copies of this server running on a number of hosts in the network. This topology is shown below:



This second topology places some restrictions on the business objects and application objects and is only appropriate if you have application objects which are the points of entry for clients. The real restrictions for this topology based on the function available in Component Broker are as follows:

- Application objects or other business objects (such as composed business objects) initiate all transactions and sessions. This means that there are no client initiated transactions or sessions.
- Each host can run one or more servers if the third tier resource managers allow multiple sets of connections from a single host.

- Each of the hosts used to scale up this topology must have access to all of the third tier resource managers because they run all of the objects. This means that in the DB2 adapter example, each host is a DB2 client talking to the DB2 server with the same permissions and authorizations.

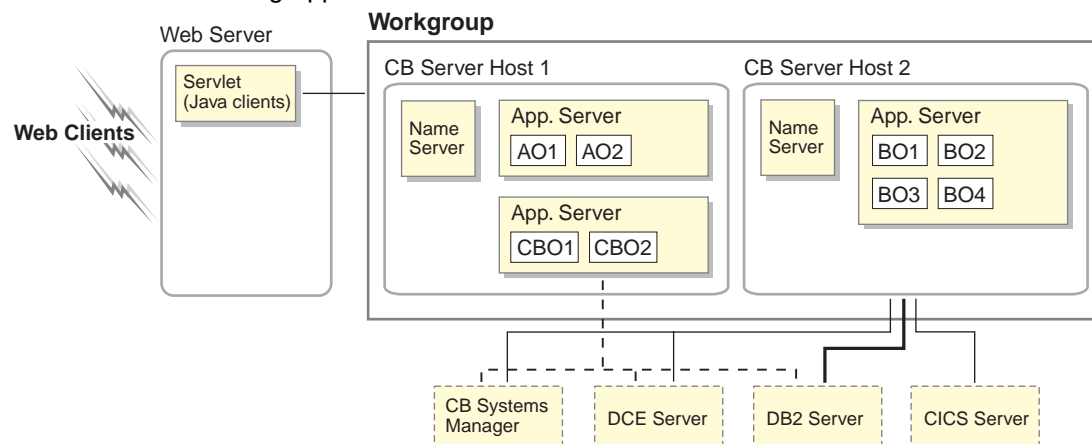
In the previous example, two application servers per host were introduced and two Component Broker server hosts were introduced as part of the workgroup. The general theme for this topology is represented by the **ao** in the above illustration. All use-cases are initiated through application objects. These control transactions and sessions ensure that the overall application conforms to the restrictions outlined above.

While this topology may have more constraints and limitations than the application information discussed earlier, it offers the most flexible horizontal scaling. Hosts and possibly additional servers per host can easily be added to the configuration when usage grows or as response times need to be adjusted. Scaling can be as simple as adding additional hosts that are clones or identical to the existing ones. Adding another processor to make a single-processor a multi-processor and subsequently adding another application server is also an easy and effective scaling tactic.

Notice also that when running with workload management there are a couple of other servers on one of the hosts in the network. The Server Group Control Point (SGCP) and Server Group Gateway (SGGW) are part of the system support required for running workload management. These should be treated as application servers from a memory usage perspective, although they will generally be a little smaller and will have different usage patterns.

## Vertical Topology

A third topology takes the first topology and uses a more pure distributed object form of scaling. This takes individual objects from the application family and places them on different servers, perhaps on different hosts throughout the network. This topology (illustrated following) has some interesting restrictions and offers some interesting opportunities.



Most importantly, the access to third tier resource managers can be isolated to the hosts which contain business objects which are backed by particular adapters. For example, if the CICS-backed objects are on host1 and the DB2 backed objects are on Component Broker servers on Component Broker host2, then only Component Broker Host1 needs access to CICS and only Component Broker Host2 needs access to DB2. This allows for more selective control over third tier resource access.

On the other hand, scaling this topology will happen vertically. This means that tuning the amount of time spent processing methods on particular objects will mean increasing the memory or processor speed of a particular host in the network. There can also be additional distribution of objects to achieve further throughput gains. Running ten distributed objects on five servers on two hosts is one variation on this third topology. It is also possible to run the same ten distributed objects on five servers on five hosts if the cost

of the additional remote method calls is less than the gain in processing power of introducing additional computers to the network.

## Advanced Topologies

More advanced topologies that mix together the horizontal scaling capabilities (those that leverage WLM) and the vertical scaling capabilities (those that rely on distribution of objects to servers on hosts) are also possible. These can be constructed by combining the topologies described above and are not covered further in this chapter.

## Additional Options

By following the basic structures of these topologies, additional tuning and configuration alterations can be considered. The first interesting alteration might be the addition of DCE replicas for naming and security. These can be added to the Component Broker server hosts in both the horizontal and vertical topologies. Using DCE replicas can increase performance, reduce bottlenecks, and improve overall system reliability. However, if using DCE replicas, the computer running the DCE replica server should be on the same platform and have similar performance characteristics as the DCE master server. Consult with your IBM representative for additional information on deploying DCE replicas.

Running in mixed host mode is another possibility to consider. It is a valid and useful option to deploy DCE on a high-performance AIX host serving Component Broker application server hosts running on Windows NT with DCE clients.

Mixed mode can also pertain to Component Broker server hosts. It is not unreasonable to have the application objects located on Windows NT Component Broker server hosts and have the business objects that talk directly to the third tier resource managers on Component Broker for AIX or OS/390 Component Broker. Windows NT to AIX communication using IIOp between distributed objects in a solution is part the basic CORBA architecture for interoperability.

If a topology that involves intranets and applets is being used, it is important to recognize that given Java VM security restrictions and the technology available in Component Broker today, the first business objects that are used from the Java applet must reside on the same computer as the Web Server from which the applet was downloaded. Follow the rules of the vertical topology (detailed previously) to get the best possible distribution and avoid having to have a huge computer as a combined web server and Component Broker Server.

Other advanced options are also possible.

## Testing the Topology

It is often not possible to do extensive modeling of the topology with the developed application before deployment. Therefore it is useful to have a prototypical set of applications. Calculations of how a prototypical application runs on the selected topology can provide a base set of numbers. These numbers are then modified based on application differences to determine the proper resource utilization and response time numbers for a chosen topology.

Component Broker does not provide a comprehensive set of numbers or a comprehensive set of reference applications. Deployment of large distributed object systems does not benefit from a large body of historical empirical data and associated analysis. There is currently no official benchmark for distributed objects, with published tables based on an agreed-to specification, and there is no certifiable TPC-C for objects.

However, there are two applications which are available. These applications are described following.

The first application is based on the install verification programs provided in the samples directory of Component Broker. The model is based on one relational database, known as the Policy2 database. The application, called AdvancedPolicyApp is actually the seed from which other volume-based applications have been created. This set of simple applications contains usages of Component Broker's features for:

- Static-SQL backed data objects.
- Caching Service based data objects (configurable for both optimistic and pessimistic).
- Query service utilization (query on homes, query evaluators, and data array queries).

The second application is a multiple database application that today only uses static SQL backed data objects. While the resource managers used are simple, the distribution opportunities are many. This configuration and capacity planning application comes with application families which are configurable to run with Workload Management (WLM) support or on multiple servers on multiple hosts following the third vertically scaleable topology described earlier.

The basic process for calculating the proper configuration and capacity requirements is as follows:

1. Select a topology which best fits the application architecture and application information described earlier.
2. Select a representative set of hardware. You can model the entire deployment configuration by using just a few computers.
3. Select the combination of the "Policy" and "Big3" applications which best fits the model of the application being deployed.
4. Optionally, modify the sample applications to better fit the deployment application. Wait times, additional database accesses, and some simple tuning can be very easily done. These sample applications are available in source code and Object Builder model form.
5. Install and Configure these applications into the test configuration. Additional client application programs can be easily created that run against the same basic business objects that are provided.
6. Run a model of the workload and gather response time and throughput numbers. This will give an indication of what the available hardware is capable of doing.
7. Adjust the response time and throughput numbers based on extrapolations from the model to the actual deployment application.
8. Calculate the hardware required based on the adjusted numbers above.

---

## An Estimation Technique

In cases where quick estimates must be made and full topology testing is not possible, another technique can be leveraged. This technique involves using known execution times of specific actions in combinations that map to the operations that a particular application might use.

The following steps will help you estimate the number of systems you will need in order to deploy your application:

1. Define the different transactions that comprise your application.
2. For each transaction, estimate the maximum number of requests that your users will make per hour.
3. For each transaction, put together a profile of the major Component Broker APIs that will be called on the server. A transaction profile consists of the list of APIs called, along with the average number of calls made to each API.

**Note:** Table 1 on page 20 is a list of the commonly used APIs, along with their execution time on the standard hardware configuration. The execution time will likely be different if you select a

different hardware configuration. To obtain an estimate of the relative throughput difference between your systems and the standard hardware configuration, consult your system provider or check a web site (for example, <http://www.spec.org>) that provides competitive benchmarking information.

- For each transaction, calculate the system load for one request by completing the following set of equations:

```

First API time = execution time of first API * average number of times called
Second API time = execution time of second API * average number of times called
.
.
.
Last API time = execution time of last API * average number of times called
-----
Total execution time for one transaction

```

- Calculate the system load for the maximum number of transaction requests per hour by completing the following set of equations:

```

First transaction time = execution time of first transaction * number of times called per hour
Second transaction time = execution time of second transaction * number of times called per hour
.
.
.
Last transaction time = execution time of last transaction * number of times called per hour
-----
Total execution time for all transactions

```

- The results from the previous step are the number of milliseconds of system time required to complete one hour's worth of user requests. Convert it to number of systems required by using the following equation:

```

number of standard systems required = result from step 5 / (1000 * 60 * 60)

```

- If you have selected a hardware configuration different than the standard hardware configuration and you have not yet made adjustments to account for this difference, now is the time to do so.

<i>Table 1. Execution Time of Commonly Used APIs</i>	
<b>API</b>	<b>Execution Time in Milliseconds</b>
IManagedClient::IHome::createFromPrimaryKeyString()	9.8
average Managed Object method request	5.6
CosTransactions::Current::begin()	39.3
CosTransactions::Current::commit()	39.3
IExtendedQuery::QueryEvaluator::evaluate_to_data_array()	73.7
IExtendedNaming::NamingContext::resolve_with_string()	142.5
IManagedClient::IHome::findByPrimaryKeyString()	5.7

## Creating the Installation Plan

After the configuration is calculated, then you can quickly decide which computers to start installing first and in what order. You can also determine which installation verification programs can be run and how to go about verifying the configuration.

Assume that you end up with the need for seven Component Broker server hosts, each of which is a P266

computer with 160 megabytes of memory as the Component Broker servers. There would also be additional computers for the DCE and Component Broker Systems Manager. You would install the following computers in the following order:

#### **DCE Server**

This would be the Cell Server and can be any computer with 128 megabytes of memory with a processor of 200 megahertz or better. This computer owns the cell. There is no Component Broker software on this computer.

#### **System Manager computer**

A computer of almost any reasonable configuration and size will be just fine. This computer does not actually get involved in each transaction. It is merely the repository for management information and the interface to viewing and changing anything that involves the Component Broker configuration. If a configuration other than the sample configuration is to be used, creating that configuration after installing this host is recommended. The Component Broker server hosts installed next rely upon this computer.

#### **Component Broker Server Host 1...N**

These are the workhorse computers. They are managed by the System Manager computer described previously. One of these computers will be the bootstrap host that clients and other servers use. This particular computer has an incremental need to do more work than any arbitrary Component Broker server host. Each of these computers will become a member of the configuration which is defined by the System Manager computer. If the sample configuration is used, very little additional work needs to be done. If a special configuration is used, then each Component Broker server host must be properly "plugged in" to the configuration to ensure proper nametree federation.

These hosts, of course, are also configured to be in the same DCE cell. They also have to be configured to access the third tier resource managers that are consistent with the topology which has been chosen.

Once all of these computers are installed, it is reasonable to install and configure something on each Component Broker server host using the System Manager computer to verify that each host is capable of doing work in this environment.

Once these computers are configured and before any applications are configured onto servers that go onto these hosts, the configuration should be activated and the name servers on each host should be verified as running. Servers can be defined and configured onto hosts before or after this first activation. Servers can actually be configured onto hosts before or after applications are configured onto the servers, although the servers should be configured onto the hosts and activated before applications are configured onto the servers. This provides a more incremental approach to building the operational configuration.

Loading the application family and configuring the applications from that family onto individual servers is easy. Just follow the plan that corresponds to the topology you have chosen. It is very important that the application family be created with the applications that are necessary to realize the topology you chose. If a topology is chosen that cannot be configured with the applications provided, then alterations to the application family or the topology are necessary.

---

## **Conclusion**

Distributed object systems offer a high degree of flexibility when it comes to configuring a deployment topology. This flexibility also comes with some complexity. This chapter has outlined the basic dimensions of a distributed application that must be understood in order to achieve a successful application deployment on Component Broker. These dimensions provide raw material for the calculation of the configuration. Each deployment configuration will have similarities, but all will be inherently different

because of the use cases being executed, the response time requirements and the nature of the existing systems that are to be leveraged along the way.

This chapter asks more questions than it provides answers. It is hoped, however, that it has provided valuable insights for administrators as they plan and execute deployment plans for Component Broker applications.



---

## Chapter 3. Planning for the Component Broker Installation

This section discusses the following information that you may find useful in planning your Component Broker installation:

- “Packaging”
- “System Requirements” on page 24
- “Prerequisites” on page 24
- “Component Broker Packages” on page 28
- “Installation Considerations” on page 32
- “Using the Prerequisite Checking Tool” on page 34

---

### Packaging

IBM Component Broker is available in the following separately orderable workstation packages:

#### Component Broker for Windows NT and AIX Connector

This package includes:

- WIN CConnector for Windows NT compact disc
- AIX CConnector for AIX compact disc
- CConnector Clients compact disc
- Supplemental Programs compact disc
- Associated documentation

#### Component Broker for Windows NT and AIX CICS and IMS Application Adaptor

The package includes:

- CICS and IMS Application Adaptor compact disc
- Associated documentation

**Note:** The CConnector package is a prerequisite.

#### Component Broker for Windows NT and AIX Oracle Application Adaptor

The package includes:

- Oracle Application Adaptor compact disc
- Associated documentation

**Note:** The CConnector package is a prerequisite.

#### Component Broker for Windows NT and AIX Toolkit

This package includes:

- Component Broker for Windows NT and AIX Connector package
- WIN CBToolkit for Windows NT compact disc
- AIX CBToolkit for AIX compact disc
- WIN IBM VisualAge C++ for Windows, Version 3.5.5 package
- WIN IBM VisualAge for Java, Enterprise Edition, Version 2.0 compact disc
- Component Broker for Windows NT and AIX CICS and IMS Application Adaptor package
- Component Broker for Windows NT and AIX Oracle Application Adaptor package
- Associated documentation

---

## System Requirements

This section lists the optimal system requirements for Component Broker.

### **WIN** Component Broker Server (run time or development)

- 200 MHz processor or better
- At least 128 MB of memory
- At least 5 GB of disk space
- 800x600 capable display
- At least 300 MB of paging space (500 MB optimal for Object Builder performance)

### **AIX** Component Broker Server (run time or development)

- 233 MHz processor or better
- At least 320 MB of memory
- At least 6 GB of disk space
- X-Server capable display
- At least 500 MB of paging space

### **WIN** Component Broker C++ or ActiveX client (run time)

- 60 MHz processor or better
- At least 32 MB of memory
- Base Operating System plus 150 MB of disk space

### **AIX** Component Broker C++ Client (run time)

- At least 64 MB of memory
- Processor and disk space as required by the native operating system.

### **Component Broker Java client**

- Processor, memory, and disk space as required by the native operating system.

---

## Prerequisites

Table 2 and Table 3 on page 26 list the products that are prerequisites for Component Broker. They are not included with Component Broker unless otherwise noted. Component Broker is only supported with the listed versions and fix levels.

Table 4 on page 26 shows which prerequisite products are required for each of several typical Component Broker machine configurations. The prerequisite products must be installed in a specific order to avoid conflicts. See “Installation Order” on page 41 for the correct order of installation.

<b>Function</b>	<b>Windows NT</b>	<b>IBM AIX</b>
1. Operating System	<ul style="list-style-type: none"><li>• Microsoft Windows NT Workstation 4.0</li><li>• Microsoft Windows NT Server 4.0</li></ul> <p>Note: Component Broker requires Microsoft Windows NT 4.00 Service Pack 3 with either operating system)</p>	IBM AIX, Version 4.2.1
2. Communications	TCP/IP (provided by the operating system)	TCP/IP (provided by the operating system)

3. Web Browser	HTML 3.2-capable and JDK 1.1.5-capable, or later (for example, Netscape Navigator 4.07 or Microsoft Internet Explorer 4.01 SP1)	HTML 3.2-capable and JDK 1.1.5-capable, or later (for example, Netscape Navigator 4.07)
4. Web Server	<ul style="list-style-type: none"> <li>• Apache v1.3.3</li> <li>• Netscape Enterprise Server Standard Edition, v3.5.1</li> <li>• Microsoft Internet Information Server v4.0</li> </ul>	<ul style="list-style-type: none"> <li>• Apache v1.3.3</li> <li>• Netscape Enterprise Server Standard Edition, v3.5.1</li> </ul>
5. Cell Directory Server	IBM DCE v2.0 server, EC05	IBM DCE v2.2 server, PTF level 2
6. Cell Directory Client	IBM DCE v2.0 server, EC05 (supplied with Component Broker)	IBM DCE v2.2 client, PTF level 2 (supplied with Component Broker)
7. Security Service Server	IBM DCE v2.0 server, EC05	IBM DCE v2.2 server, PTF level 2
8. Security Service Client	IBM DCE v2.0 server, EC05 (supplied with Component Broker)	IBM DCE v2.2 client, PTF level 2 (supplied with Component Broker)
9. Database Service Server	<ul style="list-style-type: none"> <li>• IBM DB2 Universal Database Personal Edition, Version 5.0 or 5.2</li> <li>• IBM DB2 Universal Database Workgroup Edition, Version 5.0 or 5.2</li> <li>• IBM DB2 Universal Database Enterprise Edition, Version 5.0 or 5.2</li> </ul>	<ul style="list-style-type: none"> <li>• IBM DB2 Universal Database Enterprise Edition, Version 5.0.0.20 or 5.2</li> <li>• IBM DB2 Universal Database Workgroup Edition for AIX, Version 5.0 or 5.2</li> </ul>
10. Database Service Client	IBM DB2 Universal Database Client Application Enabler, Version 5.0 or 5.2 (Version 5.0 supplied with Component Broker)	IBM DB2 Universal Database Client Application Enabler, Version 5.0 or 5.2 (Version 5.0 supplied with Component Broker)
11. Database Connection	<ul style="list-style-type: none"> <li>• IBM DB2 Connect Personal Edition</li> <li>• IBM DB2 Connect Enterprise Edition</li> </ul>	IBM DB2 Connect Enterprise Edition
12. Oracle Database Server	Oracle8 Version 8.0.4.0.0 for Windows NT with Oracle8 RDBMS patch 8.0.4.1.1 or Oracle8 RDBMS patch 8.0.4.2.0.	Oracle 8.0.5.0.0.
13. CICS External Call Interface (ECI) Client	IBM CICS Transaction Gateway for Windows NT, Version 3.0 (Supplied with Component Broker CICS and IMS Application Adaptor; also included with VisualAge Java 2.0)	IBM CICS Transaction Gateway for AIX, Version 3.0 (Supplied with Component Broker CICS and IMS Application Adaptor)
14. APPC Communications	IBM eNetwork Communications Server Version 5.0 for Windows NT	IBM Communications Server Version 5.0 for AIX
15. C++ Compiler	<ul style="list-style-type: none"> <li>• IBM VisualAge for C++ for Windows, Version 3.5.5 (supplied with Component Broker Toolkit)</li> <li>• Microsoft Visual C++, Version 5.0</li> </ul>	IBM CSet++ for AIX, Version 3.1.4 (supplied with Component Broker Toolkit)
16. Java Compiler	IBM VisualAge Java Enterprise Edition for Windows, Version 2.0 (required for CICS and IMS Application Adaptor application development; supplied with Component Broker Toolkit)	IBM VisualAge Java Enterprise Edition for AIX, Version 2.0 (required for CICS and IMS Application Adaptor application development; supplied with Component Broker Toolkit)
17. Basic Compiler	Microsoft Visual Basic, Version 4.0	
18. Java Development Kit	JavaSoft Java Development Kit 1.1.6	JavaSoft Java Development Kit 1.1.6, level a116-19980924

19. Analysis and Design Tool	Rational Software Rational Rose/C++ 98	
20. DB2 Database Toolkit	<ul style="list-style-type: none"> <li>IBM DB2 Personal Developer's Edition, Version 5.0 or 5.2</li> <li>IBM DB2 Universal Developer's Edition, Version 5.0 or 5.2</li> </ul>	IBM DB2 Universal Developer's Edition, Version 5.0 or 5.2
21. Database Design Tool	LogicWorks ERwin/ERX 3.5	
22. Application Packaging Tool	InstallShield 5.1 development kit	

The following table lists Component Broker prerequisites that are supported on the OS/2 and Windows 95 platforms.

Function	IBM OS/2	Windows 95
23. Operating System	OS/2 Warp Version 4.0	Windows 95
24. Java Development Kit	JavaSoft Java Development Kit for OS/2 1.1.6	JavaSoft Java Development Kit 1.1.6

The numbers in the following table reference rows in Table 2 on page 24 and Table 3.

Prerequisite	Application Server	System Manager	Java Client	C++ Client	ActiveX Client	System Manager User Interface	Development Workstation
Operating System	1	1	1	1 or 23	1	1	1
Communications	2	2	2	2	2	2	2 (Optional)
Web Browser (optional; required to view on-line documentation)	3	3	3	3	3 (Microsoft IE only)	3	3
Web Server	4 (Optional: required to serve Java Clients)						4 (Optional)
Cell Directory Service	5 or 6						5 or 6
Security Service	7 or 8		7 or 8 (Optional: required for DCE-based authentication on the client)	7 or 8 (Optional: required for DCE-based authentication on the client)	7 or 8 (Optional: required for DCE-based authentication on the client)		7 or 8
DB2	9, 10, or 11						9, 10, or 11
Oracle	12 (Optional)						12 (Optional)

CICS ECI Client	13 (Optional)						13 (Optional)
APPC Communications	14 (Optional)						14 (Optional)
C++ Compiler							15 (Optional: required for C++ application development)
Java Compiler							16 (Optional: required for CICS and IMS Application Adaptor application development and Java application development)
Basic Compiler							17 (Optional: required for ActiveX application development)
Java Development Kit	18		18				18 or 24 (Optional: required for Java application development)
Analysis and Design Tool							19 (Optional: recommended for "tops-down" development)
DB2 Database Toolkit							20
Database Design Tool							21 (Optional: recommended if defining / modifying database schema)
Application Packaging Tool							22 (Optional: recommended for packaging applications)

**Notes:**

1. For assistance in obtaining prerequisites not supplied in Component Broker, contact your IBM representative.
2. Component Broker is supported on and has been tested with the Java Development Kit (JDK) 1.1.6. While Component Broker will install with JDK 1.1.6 and above, these levels of the JDK are not currently supported. If you experience problems while using JDK 1.1.7 or above, you must reproduce the problems using JDK 1.1.6 before reporting the problems.

Information regarding obtaining the prerequisite level of the JDK can be found by selecting Component Broker from the IBM service web page:

<http://service.boulder.ibm.com/>

3. The IBM DB2 Connect products allow access of Component Broker data stored on DB2 for OS/390, DB2 for MVS/ESA, or DB2 for OS/400. Additional details on the DB2 packaging can be found on the Web at:

[http://www.software.ibm.com/data/db2/udb/dev1\\_ed.html](http://www.software.ibm.com/data/db2/udb/dev1_ed.html)

---

## Component Broker Packages

The Component Broker software is on the Component Broker Connector, Component Broker Supplemental Programs, CICS and IMS Application Adaptor, Oracle application adaptor, and Component Broker for Windows NT and AIX Toolkit compact discs. The CBConnector compact disc includes run-time services. The Supplemental Programs compact disc includes the prerequisite checking tool, client software, and fix packs for use with Component Broker. The CICS and IMS Application Adaptor compact disc includes the run-time services for access to CICS and IMS servers. The Oracle Application Adaptor compact disc includes the run-time services for access to Oracle servers. The CBToolkit compact disc includes application development tools and Software Developers Kits (SDKs) consisting of header and library files for use in server or client development.

### CBConnector Compact Disc

**Note:** Avoid performing a Custom Install on an individual Component Broker fileset because there is no configuration entry added to ODM to do the configuration. For a server run time installation you should always install from the CBConnector.TYPICAL LPP level.

The installable software on the CBConnector compact disc includes:

#### Component Broker Server

Install this component on a system that you want to function as an object or component server.

#### C++ Client

Install this component if you want to run Component Broker C++ CORBA client applications developed using VisualAge C++ for Windows (for Windows NT) or C Set++ (for AIX) on the system.

#### Java Client

This is similar to the C++ client, but is implemented in Java and supports Java client applications. The Java client is a CORBA client. Java clients can use the SSL-based security support provided by Component Broker.

#### ActiveX Client

Install this component if you want to run client applications that support the rendering of Component Broker objects as ActiveX/COM interfaces. This component contains a C++ client Object Request Broker.

#### System Manager

Component Broker products and applications deployed on a host can be managed by the System Manager if the System Management Agent component is installed on the host. Any host can serve as a System Manager platform if the System Manager is installed. The System Manager does not need to reside on each host and is typically executed on a limited number of hosts in an enterprise. Either the System Manager or the System Management Agent can be installed on a host, but not both.

### **System Manager User Interface**

The System Manager User Interface can be installed on any host, and needs to co-reside with a System Management Agent or System Manager. There can be multiple instances of the System Manager User Interface installed and active at any point in time. Each System Manager User Interface is connected to one System Manager.

### **System Management Agent**

Component Broker products and applications deployed on a host can be managed by the System Manager if the System Management Agent component is installed on the host. Either the System Manager or the System Management Agent can be installed on a host, but not both.

### **Application Adaptor**

The DB2 Application Adaptor provides a location for instances of managed objects. It is responsible for providing system capabilities for its managed object instances. You must select the DB2 Application Adaptor if the server is to be installed on your system.

### **Documentation**

The complete set of online documentation in HTML 3.2 format is included on the Component Broker Connector compact disc.

## **CBConnector Clients Compact Disc**

**Note:** Avoid performing a Custom Install on an individual Component Broker fileset because there is no configuration entry added to ODM to do the configuration. For a server run time installation you should always install from the CBConnector.TYPICAL LPP level.

The installable software on the CBConnector Clients compact disc includes:

### **C++ Client**

Install this component if you want to run Component Broker C++ CORBA client applications developed using VisualAge C++ for Windows (for Windows NT) or C Set++ (for AIX) on the system.

### **Java Client**

This is similar to the C++ client, but is implemented in Java and supports Java client applications. The Java client is a CORBA client. Java clients can use the SSL-based security support provided by Component Broker.

### **WIN ActiveX Client**

Install this component if you want to run client applications that support the rendering of Component Broker objects as ActiveX/COM interfaces. This component contains a C++ client Object Request Broker.

### **System Management Agent**

Component Broker products and applications deployed on a host can be managed by the System Manager if the System Management Agent component is installed on the host. Either the System Manager or the System Management Agent can be installed on a host, but not both.

## **CICS and IMS Application Adaptor Compact Disc**

The CICS and IMS Application Adaptor compact disc contains run-time environment software required to access a CICS or IMS server.

## Oracle Application Adaptor Compact Disc

The Oracle Application Adaptor compact disc contains run-time environment software required to access an Oracle database.

## Supplemental Programs Compact Disc

The Supplemental Programs compact disc contains software for the AIX, Windows 95, and Windows NT platforms. The contents of this compact disc and the directory for each are contained in the following table:

<i>Table 5. Contents of the Supplemental Programs compact disc</i>			
	<b>Windows</b>	<b>AIX</b>	<b>Comments</b>
<b>VAC++ Patch</b>	nt\fixes\vacpp	n/a	You must replace the version of cppwbe35.exe in your compiler's bin directory with this version.
<b>DB2 CAE</b>	nt\db2\db2v5cli	aix/db2/db2v5cli/db2v5cli.tar	If you do not have a DB2 server installed on this system, you must install the CAE to support Component Broker.
<b>DCE Client</b>	nt\dce\dce20cli	aix/dce2.2cli	The DCE client files contain the application developers kit, run-time environment, and tutorials for DCE, plus the required PTFs for AIX.
<b>Fix Packs</b>	nt\fixes\dce\v20_eco5	aix/fixes	The required DCE fixes.
	nt\fixes\dce\v20_eco5\patches	n/a	Replacement files (libdce.dll, libdce.dbg, dcesetup.exe, and dcesetup.dbg) to fix a CDS user cache corruption problem.
	n/a	aix/fixes	Contains fixes for: <ul style="list-style-type: none"> <li>• AIX multi-processor computer</li> <li>• DCE for Component Broker server and clients</li> <li>• C Set++ for application development on Component Broker server and C++ client</li> <li>• DB2 Version 5.0</li> </ul>
<b>Tivoli</b>	Tivoli	Tivoli	Integration with Tivoli System Management.
<b>Prereq Checker</b>	prereq\nt	prereq/aix	This tool is a stand-alone Java program that can be run to display a list of prerequisite software that exists or needs to be installed on the computer before installing Component Broker.



## CBToolkit Compact Disc

The application development tools can be installed on any computer with a configuration that includes

- The development compilers:

WIN VisualAge C++ or Microsoft Visual C++ and JDK 1.1.6.

AIX C Set++ and JDK 1.1.6.

- Access to DCE and DB2.
- A Component Broker server or client.

### Notes:

1. For assistance in obtaining an appropriate Java Development Kit, contact your IBM representative.
2. Component Broker is supported on and has been tested with JDK 1.1.6. While Component Broker will install with JDK 1.1.7 and above, these levels of the JDK are not currently supported. If you experience problems while using JDK 1.1.7 or above, you must reproduce the problems using JDK 1.1.6 before reporting the problems.

Using the application development tools, you can develop Component Broker applications on one system (for example a client) and deploy the applications on another. The Component Broker for Windows NT and AIX Toolkit compact disc includes:

### CBToolkit (Application Development Tools)

A set of visual tools used to define, implement, deploy, and debug distributed Component Broker applications.

### Server SDK

Header and library files for developing applications on the server.

### C++ Client SDK

The C++ client SDK which includes header and library files for developing applications on a Component Broker C++ client.

### WIN ActiveX Client SDK

The ActiveX client SDK which includes header and library files for developing applications on a Component Broker ActiveX client.

#### Windows NT only

If you install the ActiveX client SDK on the same system on which a C++ client is installed, you need to create a two-compiler environment. For additional information, see Appendix D, "The Two-Compiler Environment (NT only)" on page 189.

### Java Client SDK

The Java client SDK which includes header and library files for developing applications on a Component Broker Java client.

### CICS and IMS Application Adaptor SDK

Header and library files for developing applications with the Component Broker CICS and IMS Application Adaptor.

### WIN Samples

A set of run-time (pre-compiled) and buildable (source) sample applications. The samples included in this selection apply to the CBConnector package, the CBToolkit package, the CICS and IMS Application Adaptor package, and the Oracle Application Adaptor package.

## **AIX** Client Samples SDK

A set of run-time (pre-compiled) and buildable (source) sample applications. The samples included in this selection apply to the CBConnector package, the CBToolkit package, the CICS and IMS Application Adaptor package, and the Oracle Application Adaptor package.

---

## Installation Considerations

This section includes considerations for planning your installation. Considerations include:

### Installation

- Component Broker must be installed on a local drive (Windows NT) or file system (AIX). However, the install image can be accessed through a network.
- Before you install Component Broker, you need to plan your site to determine which Component Broker components to install on which computer.
- **WIN** If your system currently includes any of the following software versions, you must uninstall that version and reinstall the currently required version:
  - Component Broker Version 2.0 or earlier.
  - Any version of DCE earlier than Version 2.0.
  - Any version of DB2 earlier than Version 5.0 (client or server) if installed on a system with a Component Broker server.
  - Any version of IBM VisualAge C++ for Windows earlier than Version 3.5.5

In addition, ensure that the JavaSoft Java Development Kit (JDK) 1.1.6 is listed in the classpath before any previous versions of the JDK.

- See the “Installation and Configuration” section of the *Late Breaking News* provided with Component Broker for additional important setup information.

### DB2

- The DB2 client is included on the Supplemental Programs compact disc. The Database Server is part of a separately licensed DB2 package.
- All Component Broker servers must co-reside with DB2. Either a DB2 server or a DB2 client must be installed on the system before a Component Broker server can be installed.
- All Component Broker servers must have access to a DB2 server.
- **AIX** After installing DB2 Version 5.0 for AIX, you must install the DB2 PTF that is available on the Supplemental Programs compact disc.

## **WIN** VisualAge C++ for Windows

- VisualAge C++ is a prerequisite for server application and C++ client development environments only and is included in the CBToolkit package.
- After installing VisualAge C++, you must install the patch which is available on the Supplemental Programs compact disc.
- After installing VisualAge C++ for Windows, you must install or reinstall the Windows NT Service Pack 3.

## AIX C Set++

- C Set++ is a prerequisite for server applications and the C++ client development environments only and is included as part of the AIX operating system.
- After installing C Set++ for AIX, you must install the PTF which is available on the Supplemental Programs compact disc.

## DCE

- The DCE client and its required fix are included on the Supplemental Programs compact disc. The DCE Security and the DCE Cell Directory Server are part of a separately licensed DCE package.
- WIN DCE must be stored in a file system with a security protection option (NTFS).
- Unsecure clients cannot access secure servers.
- DCE is required for Component Broker server hosts (host machines that contain one or more Component Broker servers), and any Component Broker client hosts that are enabled to use DCE-based security.
- All Component Broker servers must have access to a DCE Security Server and a DCE Cell Directory Server for object security and naming purposes.
- You must determine which systems will be DCE Security and Naming servers. You can install DCE on a single computer or you can break it up into its component parts and install each server on separate computers.
- All Component Broker servers and clients must be installed in the same DCE cell. Provide the same DCE cell name (the realm name) when configuring your DCE servers and clients.

## TCP/IP

- TCP/IP is a prerequisite for all Component Broker components. TCP/IP is installed as part of the operating system.

## Component Broker

- Consider using the **System Manager** option on your first system.
- Consider using the **Application Server** install option on other computers that you want to be Component Broker servers.
- Bootstrap Host considerations:
  - Determine which hosts will be bootstrap hosts. A bootstrap host can be any host (with the Component Broker server installed) from which remote clients can bootstrap into the system name space.
  - Determine which bootstrap host each Component Broker computer will use.
  - Determine the port ID to use for each bootstrap host. The port ID supplied during installation must match the port ID configured with the corresponding bootstrap host in its Component Broker TCP/IP Protocol Image. The TCP/IP Protocol Image for a host can use the default System Management port ID (900). This port ID can be changed after installation if it is used by another system.
- Java considerations:
  - Component Broker is supported on and has been tested with JDK 1.1.6. While Component Broker will install with JDK 1.1.7 and above, these levels of the JDK are

not currently supported. If you experience problems while using JDK 1.1.7 or above, you must reproduce the problems using JDK 1.1.6 before reporting the problems.

- When using the Component Broker Java Client as a downloaded client, the following are the prerequisites for the browser:
  - HTML 3.2 or later
  - JDK 1.1.5 or later

The browsers specifically tested with Component Broker are:

- Microsoft Internet Explorer 4.01 SP
- Netscape Navigator 4.07

- A web server is required if Java applets, which enable Java clients, are to be downloaded from the Web. The following web servers are tested and supported for use in serving the downloaded Component Broker Java client:

- Apache 1.3.3
- Microsoft Internet Information Server (IIS) 4.0
- Netscape Enterprise Server Standard Edition 3.5.1

- Component Broker does not work with the Microsoft JRE.

- System Manager User Interface consideration:

- **AIX** Motif 4.2.1 is required to use the System Manager User Interface.

- **WIN** ActiveX client considerations:

- Microsoft Visual C++ is a prerequisite for the ActiveX client development environment only.
- Microsoft Internet Explorer, Version 3.0 is a prerequisite for the ActiveX client only.

- **AIX** Component Broker Application Installation considerations:

- Before installing a Component Broker Application on a host running a Component Broker server, ensure that there is adequate space for the application on the /var file system. The applications are installed into the /usr/lpp/CBConnector/aixApps directory. This path is a symbolic link to the /var/CBConnector/aixApps directory on the /var filesystem.

An alternative is the creation of a separate /var/CBConnector file system. With the creation of the var/CBConnector file system, there is no need to enlarge the /var file system. The creation of this file system is discussed in Chapter 9, “Installing Component Broker for AIX” on page 121.

---

## Using the Prerequisite Checking Tool

The Prerequisite Checker tool is a stand-alone Java program that can be run to display a list of prerequisite software that exists or that needs to be installed on the computer before installing Component Broker. The program, given the Component Broker configuration that is to be installed, cross checks the list of prerequisites and provides the information that the chosen configuration either:

- Will install OK.
- Will not install OK. In this case, the tool lists any prerequisites that are missing and that need to be installed.

You can run the tool in a GUI format or in a non-GUI format. When the GUI version is invoked, the tool displays a list of products and their corresponding packages. By default the **Component Broker Run Time** product is preselected and the list packages for this product are displayed.

For the non-GUI version, the tool checks all the products and their corresponding packages. The output is displayed or you can redirect the results into a logfile.

The tool runs on the AIX, Windows NT, and Windows 95 platforms and is available on the Supplemental CD in the following directories:

- `WIN` prereq\nt
- `AIX` prereq/aix

`WIN` To use the prerequisite checking tool in Windows NT or Windows 95, perform the following steps:

1. Insert the Supplemental Programs compact disc into the CD-ROM drive.
2. Make the CD-ROM drive your default drive.
3. Copy the contents of the `/prereq/nt` directory from the Supplemental CD to a local directory.
4. Run the `prereq` program to start the tool:

- Change to the directory containing the tool and run the `prereq` program. Enter:

```
prereq
```

**Note:** This command runs the prerequisite checking tool in a GUI format. To run the tool in a non-GUI format, enter:

```
prereq -nogui
```

The results are displayed. To pipe the output to a file, enter:

```
prereq -nogui [ logfile | path/logfile ]
```

Where *logfile* represents the name of an output file.

5. When the tool starts, select a product from the **Products** list. For each product selected, a list of packages are displayed. Select the packages from the **Packages** list that you intend to install and click the **Run Check** button.

6. When the tool finishes, the results are:

**GUI format** Displayed in a window. To see the details of these results in the Component Broker Log Window, click the **Show Details** button. To save the details, click the **Save Output** button.

**Non-GUI format** Displayed to the screen or redirected to a logfile. When redirected to a logfile, you can view the details by opening the logfile.

`AIX`

To use the prerequisite checking tool in AIX, perform the following steps:

1. Login as root (if not already logged in).
2. Insert the Supplemental Programs compact disc into the CD-ROM drive.
3. Mount the CD-ROM. Enter:

```
mount /cdrom
```

4. Copy the tar file `/prereq/aix/prereq.tar` from the Supplemental Programs compact disc to a local directory and from a shell prompt, enter:

```
tar -xvf /cdrom/prereq/aix/prereq.tar
```

5. Run the script file to start the tool:

- Change to the prereq directory created by the tar command in the previous step and run the script. Enter:

```
./prereq
```

**Note:** This command runs the prerequisite checking tool in a GUI format. To run the tool in a non—GUI format, enter:

```
./prereq -nogui [ logfile | path/logfile ]
```

Where *logfile* represents the name of your output file.

6. When the tool starts, select a product from the **Products** list. For each product selected, a list of packages are displayed. Select the packages from the **Packages** list that you intend to install and click the **Run Check** button.

7. When the tool finishes, the results are:

**GUI format:**

Displayed in a window. To see the details of these results in the Component Broker Log Window, click the **Show Details** button. To save the details, click the **Save Output** button.

**Non-GUI format:**

Displayed to the screen or redirected to a logfile. When redirected to a logfile, you can view the details by opening the logfile.

---

## Installing Component Broker





---

## Chapter 4. Installing the Prerequisite Software on Windows NT

This section contains information about and procedures for installing the software prerequisites, including:

- “Verifying the Windows NT Version”
- “TCP/IP Considerations”
- “User ID Considerations” on page 40
- “Installation Order” on page 41
- “Installing and Configuring DB2” on page 41
- “Installing VisualAge C++” on page 46
- “Installing VisualAge for Java” on page 47
- “Installing and Configuring DCE” on page 48

In addition, you can use the Prerequisite Checking Tool to verify that the prerequisite software is installed and configured correctly to be used by Component Broker. For information about the tool, see “Using the Prerequisite Checking Tool” on page 34.

---

### Verifying the Windows NT Version

All Component Broker components require Microsoft Windows NT 4.0 Workstation or NT 4.0 Server with Service Pack 3 installed.

To verify the level of your Windows NT system, watch for the operating system startup information that displays the NT build level and Service Pack level.

#### Notes:

1. If your computer does not have Service Pack 3 installed, and you plan to perform application development, install VisualAge C++ first, then install Service Pack 3.
2. If your computer already has Service Pack 3 installed, and you plan to perform application development, you must reinstall Service Pack 3 after installing VisualAge C++ and before installing Component Broker. Reinstalling Service Pack 3 ensures that the drivers for accessing databases is back to the correct level.
3. When installing or reinstalling Service Pack 3, a question may display asking if you want to replace newer files with older files. If this occurs, click the **No** button.

---

### TCP/IP Considerations

TCP/IP is a prerequisite of Component Broker. During TCP/IP configuration, ensure that you are configuring TCP/IP with a valid host and domain name. The notation for the host name and the domain name are case sensitive. For example, if the host name is “MYHOST” and you specify “myhost”; you are specifying an invalid host name.

If you realize that you used a host or domain name that is not valid to configure TCP/IP after installing Component Broker, you need to reconfigure TCP/IP, and then you need to reconfigure Component Broker using the configuration tool. Also, if you modify the host name or the domain name after installing Component Broker, you will need to reconfigure Component Broker using the configuration tool.

Dynamic IP Addressing (DHCP) is not supported by Component Broker.

---

## User ID Considerations

This section discusses the User ID considerations for DB2 and Windows NT.

You should install all prerequisite software and the Component Broker software under the same NT user ID. This helps to ensure that the environment variables and necessary authorities are set up so that you can compile programs and access databases.

### For a DB2 Administrator ID

The user ID you select must be acceptable as a DB2 administrator ID and must have NT Administrator rights. A valid DB2 user ID:

- Must contain eight or fewer characters.
- Can include letters, numbers, the at sign (@), the pound sign (#), and the dollar sign (\$).
- Cannot begin with IBM, SYS, SQL, or a number.
- Cannot be a DB2 reserved word (USERS, ADMINS, GUESTS, PUBLIC, or LOCAL) or an SQL reserved word as listed in the *SQL Reference*.

**Note:** DB2 tells you if you attempt to install using an invalid user ID.

### For a Windows NT Administration ID

If the workstation will be used only within a Microsoft Windows Networking Workgroup, use the NT User Manager to define the user ID and ensure that the new user ID is a member of the Administrators group.

If the workstation will be used only within a Microsoft Windows Networking Domain, use the Windows NT User Manager for Domains on the domain controller to define the global user ID and ensure that the new user ID is a member of the Domain Admins group. If the new user ID cannot be a member of the Domain Admins group, log on to the client workstation locally (using the Domain name that matches the host name of the workstation) and ensure that the domain-qualified user ID (without the domain qualifier) is a member of the local Administrators group. After this verification, log on again, but use the domain-qualified user ID.

For more information about Security in a Component Broker environment, see the “Security Administration” topic in the *Component Broker for Windows NT and AIX System Administration Guide*.

---

## Windows NT Configuration Considerations

The following considerations are vital to accurate performance of Component Broker on the Windows NT platform.

### NLS Considerations

Do not change code pages for a server after the server has been started. Doing so will result in INVALID OBJ REF errors. To change your code page, you must first shut down your servers, daemons, and the systems management graphical user interface. You can then change the code page using the LANG environment variable. When you restart, all components will be running under the new code page.

---

## Installation Order

Use the following installation order:

1. Install DB2 using the new user ID (see “User ID Considerations”).
2. Install VisualAge C++ for Windows using the same user ID. For ActiveX clients, install Microsoft Visual C++. For Java clients and Component Broker servers, install JDK 1.1.6.  
**Note:** Component Broker is supported on and has been tested with JDK 1.1.6. While Component Broker will install with JDK 1.1.7 and above, these levels of the JDK are not currently supported. If you experience problems while using JDK 1.1.7 or above, you must reproduce the problems using JDK 1.1.6 before reporting the problems.
3. Update the VisualAge C++ cppwbe35.exe file (replacement file is located on the Supplemental Programs compact disc).
4. Install NT Service Pack 3.
5. Install DCE using the same user ID.
6. Install the DCE fix pack (located on the Supplemental Programs compact disc).
7. Update the following DCE files using the files updates located on the Supplemental Programs compact disc:
  - DCE libdce.dll
  - libdce.dbg
  - dcesetup.exe
  - dcesetup.dbg
8. Install Component Broker using the same user ID.

---

## Installing and Configuring DB2

Component Broker supports the following transparent mapping between DB2 NT relational databases and Component Broker applications.

- Integration of Component Broker Object Transaction Service and Object Concurrency Control Service with DB2 transactions and locking.
- Integration of Component Broker Object Query Support with DB2 SQL query processing.
- By using the distributed services of the DB2 family, support for accessing local DB2 NT databases actually provides support for access to remote DB2 databases on AIX, OS/2, and MVS/ESA. This transparent access is achieved by configuring DB2 table aliases that define local tables whose actual implementation is provided by a remote DB2 server.
- A Database Caching service that implements algorithms and policies for optimizing the performance of DB2 databases within Component Broker. This service includes local caching to rows and columns, optimistic read and update policies, optimistic transaction policies, and integration of query and transactional access to data.

If installing the Component Broker server when there is an existing CBSRFCDB database, the existing database is not dropped during the installation because it could contain user data. In addition, the database that is normally created as part of the installation fails because the database already exists.

If this is the case, you may want to drop the database in order to clean it up prior to installation. On the DB2 server system, run the following commands to drop the database:

```
db2cmd db2
drop database CBSRFCDB
```

After dropping the database and extracting your data, either perform the Component Broker server installation (if not already done) or recreate the database following the instructions in “Installing a Component Broker Server on a DB2 Client” on page 173.

#### Notes:

1. You can implement transactional applications by using the DB2 transaction interfaces. There are limitations to this technique that include:
  - One active DB2 transaction per Component Broker process.
  - Transactional updates are confined to a single database.
2. DB2 systems use DB2 DDCS to implement distributed transactions. DB2 NT DDCS interacts only with DB2 AIX and DB2 OS/2. Therefore, Component Broker transactional updates of DB2 MVS/ESA require the presence of either a DB2 AIX DDCS or a DB2 OS/2 DDCS gateway. For additional information, see your DB2 documentation.
3. If you uninstall DB2, verify that any residual database directories are deleted before reinstalling DB2. Failure to delete these residual directories may cause errors and subsequent failure during a DB2 reinstallation.

## Installing DB2 NT

From a DB2 perspective, a Component Broker server is a database client. Therefore, the configuration is:

- Component Broker server must have access to a DB2 NT server. Either a DB2 server or a DB2 client needs to be installed on the system before a Component Broker server can be installed.
- You need to install the DB2 SDK as part of the overall installation on a system if you plan to use the system for developing Component Broker applications.

**Note:** The installation of the DB2 SDK places the include and lib paths in the System Environment Variables only. To compile programs using the DB2 SDK, you need to add the DB2 include and lib paths to the User Environment Variables.

This information provides an overview of the steps required to configure DB2 for Component Broker installation. The installation process and considerations for DB2 NT are independent of Component Broker. Therefore, you should follow the DB2 NT installation and configuration information. For additional information on DB2 configuration, see “Installing from a CD-ROM” in the *IBM DB2 Up and Running Guide*.

The installation process results in a Component Broker business object being able to make local SQL calls. The SQL calls could be routed to remote DB2 servers, such as MVS/ESA, through distributed DB2 services.

## Installing the DB2 NT Server

Unless you have an existing DB2 Server, you need to install at least one DB2 Server for all the Component Broker servers. Database Server is part of a separately licensed package and is not part of the Component Broker package.

**Note:** If DB2 will be on a system with a Component Broker server, you must uninstall any version of DB2 earlier than Version 5.0 and reinstall the current version of DB2.

It is useful to install DB2, DCE, and Component Broker on the same Windows NT account. This allows you to have access to all the Component Broker development tools without having to shut down and log on as a different user each time you switch between DB2 tools and Component Broker tools. The DB2 and Component Broker accounts need administrator authority, and you have to create the account using **Programs → Administrative Tools (Common) → User Manager**.

To install the DB2 NT server:

1. Insert the DB2 Version 5.0 server compact disc into the CD-ROM drive. Autoplay starts the setup program (or you can run Setup.exe from the Windows NT Explorer).
2. On the welcome screen, click the **Next** button to continue.
3. Select one of the following DB2 editions, based on your needs:
  - Workgroup** Can employ up to four processors on Intel platforms only; each Web user is considered to be a DB2 user; Connect Enterprise functionality is needed to work.
  - Enterprise** Can employ tiered processors with no maximum number on multiple platforms; Web users are not considered to be users; Connect Enterprise functionality is included as part of the product.
  - Client Application Enabler** The DB2 Client Application Enabler (CAE) enables a client workstation to access the DB2 server. By default, if you select the Workgroup or Enterprise options, it is not necessary to install the CAE.
- Note:** Either the Workgroup selection or the Enterprise selection is acceptable to Component Broker. See your DB2 documentation for more information.
4. Select the install directory.
5. Select Typical Install.
6. Type the user name and password for DB2 servers log on.
7. Click the **Next** button to continue.
8. When the installation has completed, restart your computer.
9. Optional, create or view the SAMPLES database to verify the install using **DB2 First Steps** menus that automatically start after restarting your computer.
10. From the Services folder (Control Panel → Services), ensure that the DB2 and the DB2 security services are set to automatic. These services are denoted as:
  - DB2 Security Server
  - DB2 - DB2

## Configuring the DB2 Server for Remote Client Access

If your Component Broker server is to be installed on the same computer as your DB2 server, no additional configuration is required. If you want DB2 clients to be able to access your DB2 server, perform the following steps:

1. Add the DB2COMM system environment variable with a value of TCPIP.
2. Configure the DB2 server to accept TCP/IP clients. Edit the x:\Winnt\system32\drivers\etc\Services file (where x:\Winnt represents the drive on which Windows NT is installed) and add entries for the connection and interrupt ports. Pick port numbers larger than 1024 that are not already used. The connection port name must be the same as the instance name. For example, add the following lines to the Services file:

```
DB2      3700/tcp
DB2I     3701/tcp
```

**Notes:**

- a. The fourth character in the DB2I port name is difficult to read in some browsers. The character is the letter “I” as in “IBM.”
  - b. Port names are case sensitive and must match the server settings exactly.
3. Open the DB2 command line processor and enter:  

```
update database manager configuration using svcname DB2
```

This updates the service name with the connection port name (DB2), which is also the instance name.
  4. Restart the computer.
  5. Log on and restart the servers. DB2 is now ready to start accepting client requests. There are some TCP/IP parameters that you might want to modify. See the *IBM DB2 Up and Running Guide* for more details.

## Installing the DB2 Client

If your Component Broker server is to be installed on a computer that does not have a DB2 server installed, you need to install the DB2 client. The DB2 client is included on the Supplemental Programs compact disc as part of the Component Broker package.

**Note:** If DB2 will be on a system with a Component Broker server, you must uninstall any version of DB2 earlier than Version 5.0 and reinstall the current version of DB2.

To install the DB2 client for DB2 5.0 Universal Database Server:

1. Insert the Supplemental Programs compact disc into the CD-ROM drive.
2. Display the contents of the compact disc.
3. Double-click the nt folder.
4. Double-click the db2 folder.
5. Double-click the db2v5cli folder to display its contents.
6. Double-click the Setup object to display the installation window.
7. Click the **Next** button to continue.
8. Optional, select components to administer remote servers.
9. Select Typical Install
10. Select the install directory.
11. When the install completes, restart your computer.

## Configuring the DB2 Client

To configure the DB2 client:

1. Configure the client to use TCP/IP to connect to the server. Edit the `x:\Winnt\system32\drivers\etc\Services` file (where `x:\Winnt` represents the drive on which Windows NT is installed) and add entries for the connection and interrupt ports. Pick port numbers larger than 1024 that are not already used. The connection port name must be the same as the instance name. For example, add the following lines to the Services file:

```
DB2      3700/tcp
DB2I     3701/tcp
```

**Notes:**

- a. The fourth character in the DB2I port name is difficult to read in some browsers. The character is the letter “I” as in “IBM.”
  - b. Port names are case sensitive and must match the server settings exactly.
2. Add the DB2COMM System environment variable with a value of TCPIP.
  3. Add the path x:\SQLLIB\HELP to the HELP User environment variable.
  4. Catalog the server node and the remote database:

- a. Open a DB2 command line processor.
- b. To catalog the server node enter:

```
catalog tcpip node db2node remote hostname server  
          service_name system systemname
```

You can choose any unused DB2 name-node in place of *db2node*, *hostname* in this example is the IP name of the DB2 NT server, *service\_name* is the connection port name name (as defined in the Services file), and *systemname* can be the name of an existing system or it can be an arbitrary name that does not already exist, for example, CBServer1.

- c. To catalog the remote database, enter:

```
catalog database sample as rsample at node db2node
```

Where *sample* represents the name of the remote database, *rsample* represents the name to which the database is being cataloged, and *db2node* represents the DB2 name-node.

**Note:** You can also perform these steps through the Client Setup application in the DB2 folder.

5. Restart your computer.
6. Start the DB2 security servers from the Services folder and set it to automatic. This service is denoted as:

DB2 security server

7. Open a DB2 command line processor and enter:

```
connect to rsample user userid using pw
```

Where *userid* represents your DB2 administrative user ID and *pw* is your administrative password.

**Note:** With only the DB2 CAE (Client Application Enabler) is installed, the DB2 command line does not provide list and select functions. If you need these functions, you need to add the DB2 SDK (from the DB2 server package) to the DB2 client installation.

8. If you have your own NT user ID on the server, you can use it to connect also. See your DB2 documentation for more details.

If you plan to use the system as a DB2 client installed to develop Component Broker application, you need to install the DB2 SDK. With version 5.0, you can install the DB2 SDK using the IBM Universal Database Server SDK compact disc.

## Configuring a DB2 TCPIP Connection to an MVS Host

You can make a TCP/IP connection to or from an MVS database using a TCP/IP connection with a DDCS gateway. To make the connection, perform the following procedure:

1. Open a DB2 command line processor and enter:

```
catalog tcpip node NodeName remote MVSHostName/MVSIPAddr server TCPServiceName/TCPPortNumber
```

Where:

*NodeName* Represents the name of a node.

*MVSHostName/MVSIPAddr*  
Represents the name or IP address of the MVS host.

*TCPServiceName/TCPPortNumber*  
Represents either the TCP service name or its port number defined in the services file. If the service name is used, its port number must be defined in the services file.

2. On the command line, enter:

```
catalog dcs database DCSDBName as MVSDBName
```

Where:

*DCSDBName* Represents any valid DB2 name.

*MVSDBName* Represents any valid DB2 name.

3. On the command line, enter:

```
catalog database DCSDBName as AliasDBName at node NodeName authentication dcs
```

Where:

*DCSDBName* Represents any valid DB2 name.

*AliasDBName* Represents any valid DB2 name.

*NodeName* Represents the name of a node.

4. On the command line, enter:

```
update dbm cfg using spm_name SPMName
```

Where:

*SPMName* Represents any valid DB2 name.

5. On the command line, enter:

```
update dbm cfg using spm_log_file_sz 1000
```

6. Refresh the db server.

7. On the command line, enter:

```
connect to AliasDBName user userid using password
```

Where:

*AliasDBName* Represents any valid DB2 name.

*userid* Represents the DB2 database user ID on the MVS host.

*password* Represents the valid password associated with the user ID.

---

## Installing VisualAge C++

IBM VisualAge C++ for Windows is required for any Component Broker server or C++ client that will be used to develop Component Broker applications.

Use the following procedure to install the IBM VisualAge C++ for Windows, Version 3.5.5 development environment.



**Note:** If Component Broker is installed prior to the installation of VisualAge C++, you must reinstall the Component Broker package after VisualAge C++ is installed.

Before installing VisualAge C++ on a drive with greater than one gigabyte capacity, check the amount of free space. (The VisualAge C++ installation program displays incorrect information about the available space.) When selecting options to install, ensure that the required space is less than the free space available. Ignore warnings that insufficient space is available and proceed with the installation.

1. Uninstall any version of VisualAge C++ for Windows that is earlier than 3.5.5.
2. Install VisualAge C++ for Windows, Version 3.5.5 according to the instructions provided with the product. You should perform a "Typical install."
3. Replace the installed version of the cppwbe35.exe file (located in the \bin directory) with the version supplied on the Supplemental Programs compact disc (the file is located in the \nt\fixes\vacpp directory).
4. When the installation completes, restart the computer.

**Note:** You must reinstall Windows NT Service Pack 3 after installing VisualAge C++ and before installing Component Broker. For information about Service Packs, see "Verifying the Windows NT Version" on page 39.

---

## Installing VisualAge for Java

The installation of VisualAge for Java overwrites the registry and project repository for any previously installed version of VisualAge for Java. All projects, packages, classes, interfaces, JAR files, applets, and applications are overwritten.

### Notes:

1. VisualAge for Java 2.0 Enterprise Edition will be installing client support for CICS connectivity. If a version of the CICS client earlier than CICS Transaction Gateway, Version 3.0 is installed, it must be uninstalled before installing VisualAge for Java 2.0. Failure to do so will create a mismatch between the CICS Client and the CICS Connector support available in VisualAge for Java 2.0.
2. If the CICS and IMS Application Adaptor SDK is going to be installed during the Toolkit Installation, you must also install VisualAge for Java. The installation of the SDK installs files in the x:\CBroker\etc\VisualAgeForJava directory for CICS and IMS Application Adaptor development. When you install the Toolkit, all files installed in the x:\CBroker\etc\VisualAgeForJava subdirectory are copied into the installation location for VisualAge for Java. These files represent specific Component Broker pieces that will be feature-installed within VisualAge for Java.

However, if VisualAge for Java has not been installed yet, these files cannot be copied. If VisualAge for Java is installed after the Toolkit installation (or reinstalled after the Toolkit installation), these files need to be manually copied into the new VisualAge for Java installation directory. You can manually copy the required files using the following command:

```
xcopy x:\CBroker\etc\VisualAgeForJava\ide\features \ide\features /S
```

To install VisualAge for Java, Enterprise Edition, Version 2.0, perform the following steps.

1. Insert the VisualAge for Java, Enterprise Edition, Version 2.0 compact disc into your CD-ROM drive.
2. Display the contents of this compact disc.
3. Double-click readme.txt and read the file. This file contains information about restrictions, preinstallation tips, repository migration steps, and other useful information. The installation of VisualAge for Java, Enterprise Edition, Version 2.0 does not automatically migrate existing repositories.

4. Double-click the Setup.exe object to display the installation window.
5. Perform a complete installation. Accept all installation defaults.
6. Restart your system.

---

## Installing and Configuring DCE

This section explains how to install DCE 2.0 and the ECO5 fix pack. There are two versions of DCE available, based on cryptographic capability:

- Data Encryption Standard (DES)
- Commercial Data Masking Facility (CDMF)

The DES version has stronger encryption and is available in the United States and Canada; the CDMF version is available in other geographic regions. The installation instructions are the same for both versions.

### Notes:

1. Migrating from DCE 1.1c to DCE 2.0 could lead to DCE GSS API errors. If your existing installation is DCE 1.1c, you must first uninstall DCE, then perform a complete (new) installation of DCE 2.0.
2. The version of DCE packaged in the Supplemental Programs compact disc is a DCE 2.0 run-time only package that allows you to install and configure a DCE client. To install and configure a DCE server, you need the IBM DCE for Windows NT version 2.0 compact disc.
3. The DCE Security and Cell Directory servers can be installed on separate computers. Each server requires the DCE run-time services to be installed first. Unlike DCE 1.1c or DCE 1.1.1, there is a single install program for installing DCE 2.0.
4. Before installing DCE 2.0, ensure that your system has Windows NT and Service Pack 3 installed.
5. DCE must be stored in a file system with a security protection option (NTFS).
6. Do not start any other process until DCE installation is complete. Starting another process could cause DCE installation to end prematurely.

## Installing the DCE 2.0 Server and Client

Perform the following procedure to install the DCE 2.0 server or the DCE 2.0 client:

1. Uninstall any previous versions of DCE:
  - a. To uninstall DCE version 1.1c, enter:

```
dinstall
```

This program stops all DCE services and removes DCE from the computer.
  - b. Manually remove the di\_rmdir.bat file.
  - c. Remove the C:\OPT\DIGITAL\DCE\BIN directory from the PATH system environment variable.
  - d. Manually remove the c:\opt directory and its subdirectories. Because some of the .dll files are in use, you must restart your computer before performing this step.

**Note:** If, after uninstalling a previous version of DCE, you restart your computer before uninstalling your previous version of Component Broker, a message displays stating that the libdce.dll file cannot be found. You can disregard this message.

2. Insert the IBM DCE for Windows NT version 2.0 compact disc into your CD-ROM drive. If your system autoloads the compact disc, skip to step 5 on page 49.

3. Display the contents of the compact disc.
4. Double-click the Setup object to display the installation window.
5. In the Choose Setup Language window, click the **Next** button to accept English as the install language.
6. In the Welcome window, click the **Next** button.
7. In the Select Components window:
  - a. Select components to install. For a DCE server installation, consider one or more of the following components:
    - DCE Runtime Services (required)
    - DCE Cell Directory Server (CDS)
    - DCE Security Server (SS)
    - Additional documentation, optional

For a DCE client installation select only the DCE Runtime Services.

**Note:** The DCE Cell Directory Server (CDS) and the DCE Security Server (SS) must be available in your cell or, if you are performing a stand-alone server installation, must be installed on your computer.

- b. Accept the default directory in the **Destination Directory** box or use the **Browse** button to specify a new destination directory. You can use the same directory as your earlier DCE release. For this document, the DCE install directory is denoted as "DCE\_PATH".
- c. Verify that there is sufficient disk space to install DCE. If not, choose another disk drive or cancel the installation to free additional disk space.
- d. Click the **Next** button.

**Notes:**

- a. Service Files component contains the debugging files, not the fix pack.
- b. During the install, the message:

```
failure renaming DCE to dce.old
```

indicates that a window is open, which points to a directory within the current DCE install. To resolve, close any such window and restart the install.

8. In the DCE Translations window, click the **Next** button (unless you need to install additional DCE Translations).
9. In the Cultural Conventions window:
  - a. Select **Do not modify LANG environment variable** (from the top of the list), regardless of which country you are in. The selection in this window is single choice, so any other selection is unchecked automatically.
  - b. Click the **Next** button.
10. In the Start Copying Files window, click the **Next** button to confirm all your choices and to begin installing the DCE files to your system.
11. Click the **Yes** button to view the README file.
12. Click the **Finish** button to restart your system.

**Note:** If you encounter any problems outside this document, see the DCE README file (README.TXT), or the documentation shipped with DCE, or visit the following Web site:  
<http://service.software.ibm.com/dssdce/winnt>.

## Reinstalling DCE 2.0 (Both Server and Client)

Perform the following procedure to reinstall DCE 2.0 over DCE 2.0:

1. Stop all DCE services and shut down DCE using DCESetup. Select **Administration** → **Stop** from the menu.
2. Insert the IBM DCE for Windows NT version 2.0 compact disc into your CD-ROM drive. If your system autoloads the compact disc, skip to step 5 on page 50.
3. Display the contents of the compact disc.
4. Double-click the Setup object to display the installation window.
5. In the Choose Setup Language window, click the **Next** button to accept English as the install language.
6. In the Welcome window, click the **Next** button.
7. In the Configuration Options window:
  - a. Select **Preserve**.
  - b. Click the **Next** button.

**Note:** If you select the **delete** option, you must reconfigure DCE after reinstalling.

8. In the Component Installation Options window:
  - a. Select **Reinstall** or **Modify**. These options preserve the DCE configuration information.
  - b. Click the **Next** button.
9. In the Select Components window:
  - a. Select one or more components to install:
    - DCE Runtime Services
    - DCE Cell Directory Server (CDS), optional for DCE client
    - DCE Security Server (SS), optional for DCE client
    - Additional documentation, optional
  - b. Accept the default directory in the **Destination Directory** box or use the **Browse** button to specify a new destination directory. You can use the same directory as your earlier DCE release. For this document, the DCE install directory is denoted as "DCE\_PATH".
  - c. Click the **Next** button.

### Notes:

- a. Service Files component contains the debugging files, not the fix pack.
- b. During the install, the message:

```
failure renaming DCE to dce.old
```

indicates that a window is open, which points to a directory within the current DCE install. To resolve, close any such window and restart the install.
- c. In the DCE Translations window, click the **Next** button (unless you need to install additional DCE Translations).
- d. In the Cultural Conventions window:
  - 1) Select **Do not modify LANG environment variable** (from the top of the list), regardless of which country you are in. The selection in this window is single choice, so any other selection is unchecked automatically.
  - 2) Click the **Next** button.

- e. In the Start Copying Files window, click the **Next** button to confirm your choices.
- f. Click the **Yes** button to view the README file.
- g. Click the **Finish** button to restart your system.

**Note:** If you encounter any problems outside this document, see the DCE README file (README.TXT), or the documentation shipped with DCE, or visit the following Web site:

<http://service.software.ibm.com/dssdce/winnt>.

## Installing the DCE Fix Pack

Perform the following procedure to install the DCE ECO5 fix pack:

1. Insert the Supplemental Programs compact disc into your CD-ROM drive.
2. Display the contents of the compact disc.
3. Double-click the nt folder to display its contents.
4. Double-click the fixes folder to display its contents.
5. Double-click the dce folder to display its contents.
6. Double-click the v20\_eco5 folder to display its contents.
7. Double-click the Setup.exe object to start the install.

**Note:** You will not see the following window if all the DCE services are stopped. If the A DCE component is in use window is displayed:

- a. Select **Continue the installation**.
- b. Click the **Next** button.

You do not need to shut down DCE; however, DCE needs to be restarted after you install the fix pack.

8. When the installation completes, click the **Finish** button.

**Note:** The fix for a CDS user cache corruption problem is available but is not included in fixpack ECO5. To apply the fix, you must replace the libdce.dll and (optionally) libdce.dbg files, located in the dce\dcelocal\bin directory on your computer. In addition, you must replace the dcesetup.exe and dcesetup.dbg files.

There are two versions of the files (CDMF and DES versions) included on the appropriate Supplemental Programs compact disc in the following locations:

```
nt\fixes\dce\v20_eco5\patches\libdce.dll
nt\fixes\dce\v20_eco5\patches\libdce.dbg
nt\fixes\dce\v20_eco5\patches\dcesetup.exe
nt\fixes\dce\v20_eco5\patches\dcesetup.dbg
```

You should replace the libdce.dll file after you have installed the DCE ECO5 fixpack. To replace the files:

1. Stop all the DCE daemons and any other products that might be using the libdce.dll file.
2. Insert the Supplemental Programs compact disc into your CD-ROM drive.
3. Display the contents of the compact disc.
4. Double-click the nt folder to display its contents.
5. Double-click the fixes folder to display its contents.
6. Double-click the dce folder to display its contents.
7. Double-click the v20\_eco5 folder to display its contents.

8. Double-click the patches folder to display its contents.
9. Copy the libdce.dll and (optionally) the libdce.dbg files to the dce\dcelocal\bin directory on your computer.
10. Restart the DCE daemons and any other products you previously stopped.

## Configuring DCE

All Component Broker server installs and all security enabled Component Broker client installs must have either the DCE client or the DCE server installed. If the DCE client is installed, an existing DCE cell with Security Server and Cell Directory Server must be accessible.

DCE requires your system to have TCP/IP active and connected to a network. If you want to configure DCE and Component Broker on a stand-alone system, configure a local network by installing the Microsoft Loopback Adaptor and configuring TCP/IP to use it. To configure this adaptor, open the Control Panel and select Network.

## Configuring the DCE Server

To configure the DCE server:

1. Start DCEsetup.

A window containing two panels is displayed. The left panel is a message panel, which displays the status of the setup process. The right panel is a table, which indicates the running status of all daemons that are available in DCE. You can use this window at any time to determine the run status of the DCE system and services.

2. From the **Configuration** menu, select **Create** → **Server**.
3. In the resulting dialog, click the **Yes** button to overwrite the existing configuration, if any. This opens the Server Configuration Wizard.
4. On the first page of the Server Configuration Wizard:
  - a. The default host name in the **Host Name** field is your computer name. Do not change this information.
  - b. The default cell name in the **Cell Name** field is in the format *hostname\_cell* where *hostname* is your computer name. Assign an appropriate cell name, if desired.
  - c. In the CDS Server group box:
    - 1) Ensure that the **CDS Server** check box is set.
    - 2) The **Master Server** radio button is set by default. Do not change this setting.
  - d. In the Security Server group box:
    - 1) Ensure that the **Security Server** check box is set.
    - 2) The **Master Server** radio button is set by default. Do not change this setting.
    - 3) In the **Principal Account Name** field, accept *cell\_admin* (the default administrator name, which is the administrator ID for the cell being configured) or type a different cell administrator (account) name. The remainder of this document assumes a principal account name of *cell\_admin*.
    - 4) In the **Principal Password** field, type a password. The remainder of this document assumes a principal password of *cell\_passwd*.
    - 5) In the **Confirm Password** field, type the password again.

**Important:** You need your cell administrator (account) name and password to administer your DCE cell. Record this information in a secure location for future reference.

- e. In the Distributed Time Service group box:
    - 1) Verify that the **Distributed Time Service** check box is set.
    - 2) Set the **DTS Global Server** radio button if you want your DCE clients to sync up the time with the server or accept the default **DTS Local Server** radio button.
  - f. Click the **Next** button to continue.
5. On the second page of the Server Configuration Wizard:
- a. The Security Master Server and the CDS Server group boxes are disabled.
  - b. If any of the clients (C++, ActiveX or Java) is located in a different LAN domain, set **The cell spans multiple LANs** check box and type the LAN name in the **Name of your LAN** field.
  - c. Disable **Integrated Login** and enable **Automatically start DCE services during system startup**.  
**Note:** Integrated Login cannot be enabled unless your NT user ID and password and your DCE user ID and password are the same.
  - d. Do not set the **GDA** check box. Enable this option only if you will be configuring a Global Directory Agent.
  - e. Click the **Next** button to start the configuration process.
6. On the Confirm Server Configuration page, click the **Finish** button to start the configuration process.
7. After the configuration is finished, verify that all services in the configured state are running by viewing the right panel of the DCE Setup window.

**Notes:**

- a. The DCE Autostart service does not display whether its status is either running or not running. This is normal.
  - b. If any of DCE processes or daemon processes are not shown as Running in the Status pane of the DCEsetup window, perform the following steps:
    - 1) Restart DCE. Enter the following commands:

```
dcecp stop.dce
dcecp start.dce
```

This may start the processes or daemons that were not running previously. If this does not fix the problem, continue with the following steps.
    - 2) Run the `dcecp show.cfg` command.
    - 3) If the running state of the DCE process is not Running, run the `dcesetup` command to reconfigure the DCE services on your computer.
8. View the Configured Hosts in the DCE Director for any Component Broker systems that are on LAN domains different from the server and that will be running Component Broker clients communicating with an application on the server.
9. Close the DCE Setup window.
10. Log in to DCE from the command line. Enter:
- ```
dce_login cell_admin cell_passwd
```

### Additional Information:

1. When performing a login on a new DCE installation, after entering your password you may see the following message:

    Password must be changed!

Changing the password is not required at this time, but the message could reappear the next time you login. To prevent the message from appearing, perform the following steps:

- a. Open the dcecp control program shell. At a command prompt, enter:

    dcecp.exe

    You must be logged in as a DCE administrator.

- b. At the dcecp command prompt, enter:

    account modify cell\_admin -pwdvalid yes

- c. End the DCE command session. Enter:

    exit

- d. Login to DCE again to verify that the message does not appear.

2. DCE and its services are started during NT startup. Complete initialization can take two or three minutes after a restart (depending on the CPU speed of the computer). When you restart NT, you should wait until the DCE services are active before attempting to start or administer Component Broker components. The DCE Setup window displays the running status of the DCE services.
3. To use DCE between any client and a Component Broker server located in a different LAN domain, you need to change your DCE server configuration to set **The cell spans multiple LANs** check box.
4. If you run a remote client with security enabled but without DCE credentials, you may be prompted to authenticate yourself to DCE. In this security dialog, enter the login name, realm name, and password. The login name is your DCE User ID, the realm name is the cell name (in the format */.../cellname*), and password is your DCE User password.
5. If you reconfigure the DCE server, you must *clobber* and reconfigure all the DCE clients connected to this server. Each time you perform *clobber/clean*, if the command is successful, exit and restart DCESETUP. This prevents the setup program from picking up any unwanted information that could remain in the cache.

## Configuring the DCE Client

To configure the DCE client:

1. Start DCEsetup.

A window containing two panels is displayed. The left panel is a message panel, which displays the status of the setup process. The right panel is a table, which indicates the running status of DCE on this computer. You can use this window at any time to determine the run status of the DCE system and services.

2. From the **Configuration** menu, select **Create** → **Full Client**.
3. In the resulting dialog, click the **Yes** button to overwrite the existing configuration, if any. This opens the Full Client Configuration Wizard.
4. On the first page of the Full Client Configuration Wizard:
  - a. The default host name in the **Host Name** field is your computer name. Do not change this information.



- b. Type the DCE cell name (the DCE realm name) of your services machine in the **Cell Name** field if it is incorrect or empty.
  - c. In the Security Master Server group box:
    - 1) In the **Security Master Server Host Name** field, type the fully-qualified host name where you configured the Security Server.
    - 2) In the **Principal Account Name** field, type `cell_admin`.
    - 3) In the **Principal Password** field, type `cell_passwd` (where `cell_passwd` represents your DCE administrator password).
  - d. If the **CDS Server is within Broadcast Range** check box is set, continue to the next step; otherwise, type the fully-qualified host name of the CDS Master Server where you configured the DCE Cell Directory Server in the **CDS Server Host Name** field.
  - e. In the Distributed Time Service group box:
    - 1) Verify that the **Distributed Time Service** check box is set.
    - 2) Set the **DTS Client** radio button.
  - f. Click the **Next** button to continue.
5. On the second page of the Full Client Configuration Wizard:
- a. In the **Security Master Server Host IP Address** field, type the IP address of the host where you configured the Security Server.
  - b. The **CDS Server** host name box is disabled.
  - c. If any of the clients (C++, ActiveX or Java) is located in a different LAN domain, set **The cell spans multiple LANs** check box and type the LAN name in the **Name of your LAN** field.
  - d. Ensure that the **Automatically sync local time with cell time** check box is not set unless the Cell Server is configured to be a DTS Global Server. If configured as a DTS Global Server, set this check box and verify the server name.
  - e. Disable **Integrated Login** and enable **Automatically start DCE services during system startup**.
 

**Note:** Integrated Login cannot be enabled unless your NT user ID and password and your DCE user ID and password are the same.
  - f. Click the **Next** button.
6. On the Confirm Full Client Configuration page, click the **Finish** button to start the configuration process.
7. After the configuration is finished, verify that all services are running by viewing the right panel of the DCE Setup window.
8. Close the DCE Setup window.
9. Log in to DCE from the command line. Enter:
- ```
dce_login cell_admin cell_passwd
```

#### Additional Information:

1. When performing a login on a new DCE installation, after entering your password you may see the following message:

Password must be changed!

Changing the password is not required at this time, but the message could reappear the next time you login. To prevent the message from appearing, perform the following steps:

- a. Open the dcecp control program shell. At a command prompt, enter:

dcecp.exe

You must be logged in as a DCE administrator.

b. At the dcecp command prompt, enter:

```
account modify cell_admin -pwdvalid yes
```

c. End the DCE command session. Enter:

```
exit
```

d. Login to DCE again to verify that the message does not appear.

2. The following DCE message could appear on the client console:

```
PID XXXX ERROR gss authentication  
msgID = 0x0DCEA113  
Can't get server ticket in gss_init_sec_ctx
```

You can ignore this message.

3. DCE and its services are started during NT startup. Complete initialization can take two or three minutes after a restart (depending on the CPU speed of the computer). When you restart NT, you should wait until the DCE services are active before attempting to start or administer Component Broker components. The DCE Setup window displays the running status of the DCE services.
4. To use DCE between any client and a Component Broker server in a different LAN domain, you need to change your DCE server configuration to set **The cell spans multiple LANs** check box.
5. If you run a remote client with security enabled but without DCE credentials, you may be prompted to authenticate yourself to DCE. In this security dialog, enter the login name, realm name, and password. The login name is your DCE User ID, the realm name is the cell name (in the format `/.../cellname`), and password is your DCE User password.
6. The maximum DCE certificate lifetime value for an authentication policy is 24 days. Do not set this value greater than 24 days.
7. If you reconfigure the DCE server, you must *clobber* and reconfigure all the DCE clients connected to this server. Each time you perform *clobber/clean*, if the command is successful, exit and restart DCESETUP. This prevents the setup program from picking up any unwanted information that could remain in the cache.

---

## Chapter 5. Installing the Prerequisite Software on AIX

This section includes the following topics:

- “Verifying the AIX Version”
- “AIX Configuration Considerations” on page 58
- “The User ID and \$HOME” on page 59
- “Installing and Configuring the Java Development Kit” on page 60
- “Installing and Configuring DB2” on page 61
- “Installing and Configuring DCE” on page 69
- “Installing C Set++” on page 76
- “Installing the Fixes for the Prerequisite Products” on page 76

In addition, there is a tool you can use to verify that the prerequisite software is installed and configured correctly to be used by Component Broker. For information about the tool, see “Using the Prerequisite Checking Tool” on page 34.

The following products are prerequisites for Component Broker for AIX.

- JavaSoft Java Development Kit (JDK) version 1.1.6.
- IBM AIX version 4.2.1
- IBM DB2 for AIX Version 5.0 with PTF or 5.2.
- IBM DCE for AIX Version 2.2 with PTF level 2.
- IBM C Set++, Version 3.1.4 with PTF

To complete the instructions in this chapter, you must be logged in as root (unless otherwise specified). Additionally, the instructions specify the installation, configuration, and uninstallation using the ASCII version of SMIT (using the `smitty` command from the shell prompt). To use the GUI version of SMIT, use the `smit` command from the shell prompt and modify the instructions as appropriate.

**Note:** See the “Installation and Configuration” section of the *Late Breaking News* provided with Component Broker for additional important setup information.

---

### Verifying the AIX Version

Component Broker requires AIX 4.2.1. Ensure that the machine on which Component Broker is installed is at this level. To check the level, enter `oslevel` at the command prompt. If you need to upgrade your machine to AIX 4.2.1, contact your IBM service representative.

#### Notes:

1. Component Broker for AIX requires a complete installation of AIX 4.2.1. If during the installation of Component Broker you receive any prerequisite file set notices, install these file sets from the AIX 4.2.1 compact discs.
2. The System Manager User Interface for Component Broker for AIX requires Motif to be installed on the system prior to installing Component Broker. Verify that motif is installed by entering:

```
ls|pp -l | grep X11.motif.mwm
```

If motif is installed, you should see the following output:

```
X11.motif.mwm      4.2.1.0  COMMITTED
```

If you do not see the X11.motif.mwm 4.2.1.0 file set, install it from the AIX 4.2.1 installation compact discs.

For any Component Broker installation, you must apply the following AIX PTFs in the following order:

1. The AIX PTF located on the Supplemental Programs compact disc in the aix/fixes directory.
2. The AIX PTF located on the Supplemental Programs compact disc in the aix/dce2.2cli directory.

After installing the prerequisite products, you must repeat step 1 (above). This installs additional PTFs that could not be installed previously.

For installation instructions, see “Installing the Fixes for the Prerequisite Products” on page 76.

---

## AIX Configuration Considerations

The following considerations are vital to accurate performance of Component Broker on the AIX platform.

### Paging Space

The paging space should be set to at least the same amount as the physical memory. You may want to increase the paging space above that amount if your machine does not have a lot of physical memory.

### /tmp and /var File Systems

The /tmp and /var file systems should be defined to a minimum of 60 MB.

### NLS Considerations

Do not change code pages for a server after the server has been started. Doing so will result in INVALID OBJ REF errors. To change your code page, you must first shut down your servers, daemons, and the systems management graphical user interface. You can then change the code page using the LANG environment variable. When you restart, all components will be running under the new code page.

### Component Broker File System

You should create a file system for Component Broker before you install. This file system will be used by the Component Broker run time. This file system is required only for Component Broker server machines. Perform the following steps:

1. Log in as root (if not already logged in).
2. Enter:  

```
smitty jfs
```

The Journaled File Systems screen is displayed.
3. On this screen, select **Add a Journaled File System**. The Add a Journaled File System screen is displayed.
4. On this screen, select **Add a Standard Journaled File System**. The Volume Group Name window is displayed.
5. In this window, select the volume group and press the **Enter** key. The Add a Standard Journaled File System screen is displayed.
6. On this screen:
  - a. Set the **SIZE** of file systems to 131672 512-byte blocks.
  - b. Set the **MOUNT POINT** to /var/CBConnector.
  - c. Set the **Mount AUTOMATICALLY at system restart?** to Yes.

Press the **Enter** key to create the /var/CBConnector file system.

7. Press the **F10** key to exit SMIT.

After creating the file system, you need to mount it. To mount this journaled file system, enter:

```
mount /var/CBConnector
```

**Note:** The file system may have to be increased when applications are added using **Load Application** through System Management. When the applications are loaded, the application libraries are copied to the `/var/CBConnector/aixApps` directory. If there is not enough space on `/var/CBConnector`, it will have to be increased using `smitty chjfs`.

---

## TCP/IP Considerations

TCP/IP is a prerequisite for Component Broker. During TCP/IP configuration, ensure that you are configuring TCP/IP with a fully-qualified host name that includes both the host name and the domain name. The notation for the host name and the domain name are case sensitive. For example, if the host name is `myhost` and your domain name is `austin.ibm.com`, the hostname should be specified as `myhost.austin.ibm.com` when configuring TCP/IP.

If you realize that you configured TCP/IP with an invalid host or domain name, after installing Component Broker, you need to:

1. Re-configure TCP/IP.
2. Re-configure DCE.
3. Re-configure the Component Broker run time.

In addition, if you modify the host name or the domain name after installing Component Broker, you must perform the same re-configuration steps.

---

## The User ID and \$HOME

For simplicity, create a user ID and use the same user ID for configuring DB2 and Component Broker. For the remainder of this document, it is assumed that the user ID is “`cbuser`.” If you select another user ID, modify the instructions as appropriate.

This document uses `cbuser` to configure DB2. The DB2 instance files will be created in the home directory of `cbuser`. Therefore, the file system where the home directory of `cbuser` is should be increased to a minimum of 120 MB. (During certain test cases, the database instance can grow to over 100 MB.) This file system size assumes that there are no other users or systems consuming this space. If there are other users or system in `/home`, compensate as needed.

**Note:** Once a file system is increased, it cannot be reduced.

To avoid having 120 MB as the `/home` file system, you can create a journaled file system that can serve as `/home` file system for user `cbuser`. To create this journaled file system:

1. Log in as root (if not already logged in).
2. Enter `smitty jfs`. The Journaled File Systems screen is displayed.
3. On this screen, select **Add a Journaled File System**. The Add a Journaled File System screen is displayed.
4. On this screen, select **Add a Standard Journaled File System**. The Volume Group Name window is displayed.
5. In this window, select the volume group and press the **Enter** key. The Add a Standard Journaled File System screen is displayed.
6. On this screen:
  - Set the **SIZE of file systems** to 524888 512-byte blocks.

- Set the **MOUNT POINT** to `/cbfs`.
- Set the **Mount AUTOMATICALLY at system restart?** to `yes`.

Press the **Enter** key. The `/cbfs` file system is created. Exit SMIT.

After creating the `/cbfs` file system, it needs to be mounted. To mount this journaled file system, enter:

```
mount /cbfs
```

With this file system mounted, user `cbuser` can be created and assigned this new home.

To create user `cbuser` and its initial password, enter:

```
mkuser home=/cbfs/cbuser cbuser
passwd cbuser
```

The system prompts you for the initial system password for `cbuser`.

**Note:** The first time the `cbuser` user logs in, the system prompts for the password to be changed.

## Installing and Configuring the Java Development Kit

The Java Development Kit (JDK) is required for applications that use Java Business Objects or for Java clients. The main JDK web site can provide additional information:

```
http://www.ibm.com/Java
```

To determine if the JDK is installed on your system, run the following command:

```
ls|pp -l | grep -i java
```

The following line should be displayed if it is installed on your machine:

```
Java.rte.bin      1.1.6.0  COMMITTED  Java Runtime Environment
```

If the JDK is not installed on your machine, you can obtain it from the following web site:

```
http://www.ibm.com/Java/jdk/download/
```

The 1.1.6 version (September 24, 1998) is the version that is supported by Component Broker. If this version is not available on the above web sites, contact your IBM representative.

**Note:** Component Broker is supported on and has been tested with JDK 1.1.6. While Component Broker will install with JDK 1.1.7 and above, these levels of the JDK are not currently supported. If you experience problems while using JDK 1.1.7 or above, you must reproduce the problems using JDK 1.1.6 before reporting the problems.

Once the JDK is installed on your machine, verify the version by executing the following command:

```
/usr/jdk_base/bin/java -fullversion
```

The output should be similar to the following:

```
java full version "JDK 1.1.6 IBM build a116-19980924 (JIT enabled)"
```

The `CLASSPATH` and `PATH` for your Component Broker user ID must be updated in order to use the JDK. The following lines should be added to the `.profile` of the `cbuser` user ID:

```
PATH=/usr/jdk_base/bin:$PATH
CLASSPATH=/usr/jdk_base/lib/classes.zip:.$CLASSPATH
```

---

## Installing and Configuring DB2

This section describes the steps necessary to install and configure DB2 Version 5.0 or Version 5.2. If you are developing applications that use DB2, you need the Developer's Edition of DB2. See "Prerequisites" on page 24 for more information on the supported versions of DB2.

For more information on DB2, you can visit the following Web site:

[http://www.software.ibm.com/data/db2/udb/dev1\\_ed.html](http://www.software.ibm.com/data/db2/udb/dev1_ed.html)

If you had databases previously installed on your machine at version 2.1, you must either migrate your databases to the new version of DB2 or remove your old databases and recreate them.

When migrating to the new version of DB2 there are tasks that must be done before and after the installation. These tasks are documented in the *IBM DB2 Universal Database for UNIX Quick Beginnings Version 5*. Follow the steps in the "Migrating from Previous Versions" topic to migrate the database instance and the databases within that instance to the new version of DB2.

If you choose to remove your old databases and re-create them, there are several steps you must follow. You must drop your databases and your database instances, uninstall the old version of DB2 and then install the new version of DB2.

If you are using DB2 Version 5.0, a PTF is required to operate with Component Broker. This PTF is available to you on the Supplemental Programs compact disc. If you are migrating your databases, you must perform the migration steps mentioned above to migrate to the UDB version of the database. You must then follow the steps in the `fixpack.readme` in the `aix/fixes` directory of the Supplemental Programs compact disc. This guides you through the migration of your databases to DB2 5.0.0.20.

If you are using DB2 Version 5.0 and you are creating your databases for the first time or if you are removing and recreating your existing databases, you should apply the PTF on the Supplemental Programs compact disc before configuring DB2. This avoids the migration steps required with the DB2 PTF. If you apply the PTF after you have configured DB2, you will have to follow the migration steps in the `fixpack.readme` on the Supplemental Programs compact disc.

For instructions on installing the DB2 Version 5.0 PTF, see "Installing the Fixes for the Prerequisite Products" on page 76.

## Installing the DB2 Server

The steps in the following sections were taken from the "Other Methods to Install DB2 for AIX" topic in the *IBM DB2 Universal Database for UNIX Quick Beginnings Version 5*. For additional information, refer to that topic and to the "Installing and Configuring DB2" topic in the same book.

Installing the DB2 Server is a two step process.

1. Install the bundle definitions
2. Install the software

## Step 1: Installing the Bundle Definitions

To install the bundle definitions:

1. If you are not already logged in, log in as root.
2. Insert the DB2 Universal Database compact disc into your CD-ROM drive.
3. From a shell prompt, enter:  

```
smitty easy_install_bundle
```

The Install Software Bundle (Easy Install) screen is displayed.
4. On this screen:
  - a. Press the **F4** key to receive a prompt for the input device.
  - b. Highlight the CD-ROM drive containing the DB2 compact disc and press the **Enter** button. The Select Fileset Bundle screen is displayed.
5. On this screen, highlight Media-Defined and press the **Enter** key to continue.
6. Press the **Enter** key. The ARE YOU SURE? screen is displayed.
7. Press the **Enter** key to begin the install.
8. Press the **F10** key to exit SMIT and return to the shell prompt.

## Step 2: Installing the Software

To install the software for DB2, which uses the bundle definitions:

1. If you are not already logged in, log in as root.
2. From the shell prompt, enter:  

```
smitty easy_install_bundle
```

The Install Software Bundle (Easy Install) screen is displayed.
3. On this screen:
  - a. Press the **F4** key to receive a prompt for the input device.
  - b. Highlight the CD-ROM drive containing the DB2 compact disc and press the **Enter** key to continue. The Select Fileset Bundle screen is displayed.
4. On this screen and depending of your needs, highlight one of the following DB2 bundles to install and press the **Enter** key to select it.  

```
DB2V5-UDB-Enterprise  
DB2V5-UDB-WorkGroup
```
5. Press the **Enter** key to continue. The ARE YOU SURE? screen is displayed.
6. Press the **Enter** key to begin the install.
7. Press the **F10** key to exit SMIT and return to the shell prompt.



## Installing the DB2 Software Developer's Kit

If you will be developing Component Broker applications that use DB2, you should install the DB2 Software Developer's Kit (SDK) at this time. The following procedure presents one method of installing the SDK. For another method of installation and for additional information see the "Installing and Configuring DB2" topic in the *IBM DB2 Universal Database for UNIX Quick Beginnings Version 5*.

Installing the SDK is a two step process:

1. Install the bundle definitions
2. Install the software

### Step 1: Installing the Bundle Definitions

To install the bundle definitions:

1. If you are not already logged in, log in as root.
2. Insert the DB2 Universal Database compact disc into your CD-ROM drive.
3. If you do not have a defined CD-ROM file system, see the section "Creating a CD-ROM File System" on page 76 for instructions on how to do this. These instructions assume that the CD-ROM file system is mounted as /cdrom. Mount the CD-ROM file system:

```
mount /cdrom
```

4. Change to the AIX installation directory on the CD-ROM:

```
cd /cdrom/db2/aix/db2
```

5. From a shell prompt, enter:

```
smitty easy_install_bundle
```

The Install Software Bundle (Easy Install) screen is displayed.

6. On this screen, highlight Media-Defined and press the **Enter** key to continue.
7. Press the **Enter** key. The ARE YOU SURE? screen is displayed.
8. Press the Enter key to begin the install.
9. Press the **F10** key to exit SMIT and return to the shell prompt.

### Step 2: Installing the Software

To install the SDK for DB2, which uses the bundle definitions:

1. If you are not already logged in, log in as root.
2. If the /cdrom/db2/aix/db2 directory is not your current directory on the CD-ROM, change to that directory:

```
cd /cdrom/db2/aix/db2
```

3. From the shell prompt, enter:

```
smitty easy_install_bundle
```

The Install Software Bundle (Easy Install) screen is displayed.

4. On this screen:
  - a. Press the **F4** key to receive a prompt for the input device.
  - b. Highlight the CD-ROM drive containing the DB2 compact disc and press the **Enter** key to continue. The Select Fileset Bundle screen is displayed.

5. On this screen, highlight DB2V5-SDK and press the **Enter** key to continue.
6. Press the **Enter** key to continue. The ARE YOU SURE? screen is displayed.
7. Press the **Enter** key to begin the install.
8. Press the **F10** to exit SMIT and return to the shell prompt.

## Configuring DB2 Server

If you are installing DB2 Version 5.0 for the first time or you are reconfiguring your server, you should install the DB2 PTF before continuing. Refer to the instructions in “Installing the Fixes for the Prerequisite Products” on page 76. If this is a new installation, it is not necessary to migrate your database after installing the DB2 PTF.

If you are migrating your databases from a previous version of DB2, you do not need to perform the following configuration steps. These steps were done when you first installed and configured DB2. But, if you install the DB2 Version 5.0 PTF, you must perform the steps documented in the `aix/fixes/fixpack.readme` file on the Supplemental Programs compact disc. These steps are necessary to migrate your database to the new version of DB2 after the PTF is applied.

To configure DB2:

1. If you are not already logged in, log in as root.
2. Change directory to `/usr/lpp/db2_05_00/cfg`.
3. Create library and include links. Enter:

```
./db2ln
```

This command creates symbolic links for the DB2 libraries and headers in `/usr/lib` and `/usr/include` respectively.

4. Create a new group (cbadmin). Enter:

```
mkgroup cbadmin
```

This group becomes the system administration group for the database instance.

5. Make cbadmin the primary group for *cbuser*.

```
chuser pgrp=cbadmin groups='cbadmin,staff' cbuser
```

**Note:** If user *cbuser* is not created, create this user ID as explained in “The User ID and \$HOME” on page 59.

6. Change directory to `/usr/lpp/db2_05_00/instance`.

7. Create a database instance. Enter:

```
./db2icrt -u cbuser cbuser
```

This command creates an instance by:

- Assigning a name to the instance.
- Copying and customizing the files required to run the instance into `INSTHOME/sql1ib`.
- Establishing the primary group of the instance as the system administration group (cbadmin).
- Adding the instance to the list of instances on the system.

**Note:** To prevent data loss if an instance is deleted, do not add files or directories under `$HOME/sql1ib`. Where `$HOME` refers to the home directory for *cbuser*.

8. Log out as root and log in as user *cbuser*.

### Important

You must source the file `sqllib/db2profile` or add the statements in this file to the file `.profile` of user `cbuser`. To source `db2profile`, enter:

```
. $HOME/sqllib/db2profile
```

## Setting up the DB2 License

Refer to the “Getting Started with the DB2 Universal Database” topic in the *IBM DB2 Universal Database for UNIX Quick Beginnings Version 5* for information on the DB2 license. In particular, it contains information on upgrading from a Try-and-Buy license.

## Configuring the DB2 Server for Remote Client Access

If your Component Broker server is to be installed on the same computer as your DB2 server, no additional configuration is required. If you want a separate DB2 server, the Component Broker server will act as the DB2 client. You will need to configure your DB2 server to allow remote clients to access the database by performing the following procedure.

1. As the root user, configure the client to use TCP/IP to connect to the server. Edit the `/etc/services` file and add entries for the connection and interrupt ports. Pick port numbers larger than 1024 that are not already used. The same numbers should be used on both the server and the client. The connection port name must be the same as the instance name. For example, add the following lines to the `services` file:

```
DB2    3700/tcp
DB2I   3701/tcp
```

### Notes:

- a. The fourth character in the `DB2I` port name is difficult to read in some browsers. The character is the letter “I” as in “IBM.”
  - b. The port names are case sensitive and must match the server settings exactly.
2. As your DB2 user ID, add the `DB2COMM` System environment variable with a value of `TCPIP` to `$HOME/sqllib/db2profile`. Add the following line to `db2profile`:

```
export DB2COMM=TCPIP
```

3. As your DB2 *userid*, enter the following on the command line:

```
db2 update database manager configuration using svcname DB2
```

This command updates the service name with the connection port name. The connection port name must match the name used in 1. In this example, `DB2` is used as the connection port name.

4. As your DB2 *userid*, enter:

```
db2stop
db2start
```

These commands stop and restart DB2 so that your changes take effect.

DB2 is now ready to start accepting client requests. There are some TCP/IP parameters that you may want to modify. See your DB2 product documentation for additional details.

## Installing the DB2 Client

To install the DB2 client software from the Supplemental Programs compact disc, you need to mount a file system, create a temporary file system, and untar the DB2 client files. After installing the bundle definitions and software, you need to remove the temporary file system.

### Mounting and Creating the File Systems

1. If you are not already logged in, log in as root.
2. Insert the Supplemental Programs compact disc into the CD-ROM drive.
3. Mount the file system. To mount, enter:

```
mount /cdrom
```

4. Create a temporary file system to use during the installation and change directory to that file system.

```
crfs -v jfs -g rootvg -a size=155000 -m /db2v5cli  
mount /db2v5cli  
cd /db2v5cli
```

5. From a shell prompt, enter:

```
tar -xvf /cdrom/aix/db2/db2v5cli/db2v5cli.tar
```

Now you are ready to install the DB2 client. Installing the DB2 client is a two-step process:

1. Install the bundle definitions
2. Install the software

**Step 1: Installing the Bundle Definitions:** To install the bundle definitions:

1. If you are not already logged in, log in as root.
2. From a shell prompt, enter:

```
smitty easy_install_bundle
```

The Install Software Bundle (Easy Install) screen is displayed.

3. On this screen:
  - a. Type `/db2v5cli/db2/aix/db2` as the input device.
  - b. Press the **Enter** key to continue. The Select Fileset Bundle screen is displayed.
4. On this screen:
  - a. Select Media-Defined
  - b. Press the **Enter** key to continue.
5. Press the **Enter** key. The ARE YOU SURE? screen is displayed.
6. Press the **Enter** key to begin the install.
7. Press the **F10** key to exit SMIT and return to the shell prompt.

**Step 2: Installing the Software:** To install the software for the DB2 client, which uses the bundle definitions:

1. If you are not already logged in, log in as root.
2. From the shell prompt, enter:

```
smitty easy_install_bundle
```

The Install Software Bundle (Easy Install) screen is displayed.

3. On this screen:
  - a. Type `/db2v5c1i/db2/aix/db2` as the input device.
  - b. Press the **Enter** key to continue. The Select Fileset Bundle screen is displayed.
4. On this screen, select DB2V5-Client.
5. Press the **Enter** key to continue.
6. Press the **Enter** key on the Install Software Bundle (Easy Install) screen. The ARE YOU SURE? screen is displayed.
7. Press the **Enter** key to begin the install.
8. Press the **F10** key to exit SMIT and return to the shell prompt.

**Removing the Temporary File System:** After installing the bundle definitions and software, you can remove the temporary file system `/db2c5cli`. To remove this file system, perform the following procedure.

1. If you are not already logged in, log in as root.
2. Ensure that you are not in the temporary file system. Change directory to the root directory. From a command prompt, enter:

```
cd /
```

3. Unmount the file system. From a command prompt, enter:

```
umount /db2v5c1i
```

4. Remove the temporary file system. From a command prompt, enter:

```
rmfs -r /db2v5c1i
```

5. Unmount the CD-ROM to eject the Supplemental Programs compact disc from the CD-ROM drive. From a command prompt, enter:

```
umount /cdrom
```

## Installing the DB2 Software Developer's Kit

If you will be developing Component Broker applications that use DB2, you should install the DB2 Software Developer's Kit (SDK) at this time. For more information, see "Installing the DB2 Software Developer's Kit" on page 63.

## Configuring the DB2 Client

If you are installing DB2 Version 5.0 for the first time or you are reconfiguring your client, you should install the DB2 PTF before continuing. Refer to the instructions in "Installing the Fixes for the Prerequisite Products" on page 76. If this is a new installation, it is not necessary to migrate your database after installing the DB2 PTF.

If you are migrating your databases from a previous version of DB2, you do not need to perform the following configuration steps. These steps were done when you first installed and configured DB2. But, if you install the DB2 Version 5.0 PTF, you must perform the steps documented in the `aix/fixes/fixpack.readme` file on the Supplemental Programs compact disc. These steps are necessary to migrate your database to the new version of DB2 after the PTF is applied.

Before configuring the DB2 client, you need to setup your DB2 environment and create a database instance. These steps are similar to the steps performed to set up the server.

To configure the DB2:

1. If you are not already logged in, log in as root.
2. Change directory to the `/usr/lpp/db2_05_00/cfg` directory.

```
cd /usr/lpp/db2_05_00/cfg
```

3. Create a library and include links.

```
./db2ln
```

This command creates symbolic links for the DB2 libraries and headers in the `/usr/lib` and `/usr/include` directories respectively.

4. Create a new group (cbadmin).

```
mkgroup cbadmin
```

This group becomes the system administration group for the database instance.

5. Make cbadmin the primary group for cbuser.

```
chuser pgrp=cbadmin groups='cbadmin,staff' cbuser
```

**Note:** If cbuser is not created, create this user ID as explained in “The User ID and \$HOME” on page 59.

6. Change directory to the `/usr/lpp/db2_05_00/instance` directory.

```
cd /usr/lpp/db2_05_00/instance
```

7. Create a database instance.

```
./db2icrt cbuser
```

This command creates an instance by:

- Assigning a name to the instance.
- Copying and customizing the files required to run the instance into the `INSTHOME/sql1lib` directory.
- Establishing the primary group of the instance as the system administration group (cbadmin).
- Adding the instance to the list of instances on the system.

**Note:** To prevent data loss if an instance is deleted, do not add files or directories under `$HOME/sql1lib`. Where `$HOME` refers to the home directory for *cbuser*.

### Important

You must source the file `sql1lib/db2profile` or add the statements in this file to the file `.profile` of user *cbuser*. To source `db2profile`, enter:

```
. $HOME/sql1lib/db2profile
```

## Additional Configuration for Communication with the Server

Perform the following steps to set up the communications with the DB2 server.

1. As the root user, configure the client to use TCP/IP to connect to the server. Edit the `/etc/services` file and add entries for the connection and interrupt ports. Pick port numbers larger than 1024 that are not already used. The same numbers should be used on both the server and the client. The connection port name must be the same as the instance name. For example, add the following lines to the `services` file:

```
DB2 3700/tcp
DB2I 3701/tcp
```

**Notes:**

- a. The fourth character in the DB2I port name is difficult to read in some browsers. The character is the letter “I” as in “IBM.”
  - b. The port names are case sensitive and must match the server settings exactly.
2. As your DB2 user ID, add the DB2COMM System environment variable with a value of TCPIP to \$HOME/sql/lib/db2profile. Add the following line to db2profile:

```
export DB2COMM=TCPIP
```

3. As your DB2 user ID, catalog the server node and the remote database:

- a. To catalog the server node:

```
db2 catalog tcpip node db2node remote hostname server service_name
```

You can choose any unused DB2 name-node in place of *db2node*, *hostname* in this example is the IP name of the DB2 server, and *service\_name* is the connection port name (as defined in the services file).

- b. If you already created a database on the DB2 server, you can now catalog that database on your client. In the following example, the database sample was created on the DB2 server. Enter the following command to catalog the remote database sample:

```
db2 catalog database sample as rsample at node db2node
```

Where *sample* represents the name of the remote database, *rsample* represents the name to which the database is being cataloged, and *db2node* represents the DB2 name-node.

4. Restart your client computer.
5. To test the connections, logon to your DB2 client as your DB2 *userid*. Open a command line processor and enter:

```
db2 connect to rsample user userid using pw
```

Where *rsample* is your cataloged remote database, *userid* represents your DB2 administrative user ID and *pw* is your administrative password.

6. To end the database connection, enter:

```
db2 connect reset
```

---

## Installing and Configuring DCE

## Important

Before installing DCE 2.2 (either client or server), ensure that the `bos.rte.libpthreads` (the `libpthread` library) is at the following level:

**For AIX 4.2** `bos.rte.libpthreads 4.2.1.1`

To check your `bos.rte.libpthreads` version, run the following command:

```
lslpp -l bos.rte.libpthreads
```

If the `bos.rte.libpthreads` is not at the correct level, apply all PTFs using the `smitty update_all` command. The PTFs are located on the Supplemental Programs compact disc in the `aix/dce2.2cli` directory or on the DCE Server installation compact disc.

You may see a requisite failure message stating that you cannot install the `bos.*` filesets for AIX version 4.3. You can ignore the message and continue with the installation.

After installing the PTFs, you must reboot your system using the `shutdown -Fr` command.

For additional information about installing the PTFs, see "Installing the PTFs" on page 77

For AIX, the supported DCE version is 2.2 with PTF level 2. Before installing DCE:

1. Ensure that there is no DCE installation on the machine. If there is, unconfigure and remove the earlier installation. To use DCE with Component Broker for AIX, at least one machine needs to be configured as Security Server and Cell Directory Service (CDS) Server. Additional machines can be configured as DCE Clients by installing DCE Client components only.

**Note:** For detailed information on the differences between DCE Client and DCE Server, see *Directory and Security Server for AIX, Up and Running, Version 4*.

2. If the DCE product contains README files or other instructions, read them.
3. You should create a `/var/dce` file system with a minimum of 50 megabytes before you install DCE. To create the file system, perform the following steps:
  - a. Log in as root (if not already logged in).
  - b. Enter:

```
smitty jfs
```

The Journaled File Systems screen is displayed.
  - c. On this screen, select **Add a Journaled File System**. The Add a Journaled File System screen is displayed.
  - d. On this screen, select **Add a Standard Journaled File System**. The Volume Group Name window is displayed.
  - e. In this window, select the volume group and press the **Enter** key. The Add a Standard Journaled File System screen is displayed.
  - f. On this screen:
    - 1) Set the **SIZE** of file systems to 100000 512-byte blocks.
    - 2) Set the **MOUNT POINT** to `/var/dce`.
    - 3) Set the **Mount AUTOMATICALLY at system restart?** to Yes.Press the **Enter** key to create the `/var/dce` file system.
  - g. Press the **F10** key to exit SMIT.



After creating the file system, you need to mount it. To mount this journaled file system, enter:

```
mount /var/dce
```

## Installing and Configuring the DCE Server

This section contains information on installing and configuring the DCE Server. Typically, the Component Broker server is installed on a DCE server system.

### Installing the DCE Server

To install the server software:

1. Log in as root (if not already logged in).
2. Insert the DCE compact disc into the CD-ROM drive.
3. At a shell prompt, enter:

```
smitty install_latest
```

The Install and Update from LATEST Available Software screen is displayed.

4. On this screen:
  - a. In the **INPUT device / directory for software** field, press the **F4** key to display a list of available devices of which you can select the appropriate input device (/dev/cd0) and press the **Enter** key. This screen is redisplayed with additional fields.
  - b. In the **SOFTWARE to install** field, press the **F4** key to display a list of the software packages contained on the compact disc.
  - c. In the SOFTWARE to install list, scroll down and select:
    - dce.cds
    - dce.client
    - dce.compat
    - dce.pthreads
    - dce.security
    - dce.tools
  - Note:** You can optionally install the documentation.
  - d. Press the **Enter** key. The Install and Update from LATEST Available Software screen is redisplayed, but the software packages selected are in the **SOFTWARE to install** field.
  - e. Ensure that the **AUTOMATICALLY install requisite software?** and **EXTEND file systems if space needed?** fields are yes.
  - f. Press the **Enter** key. The ARE YOU SURE? screen is displayed.
5. On the ARE YOU SURE? screen, press the **Enter** key to continue.

#### Notes:

- a. If any prerequisite failure messages are displayed, install the prerequisite software and reinstall the failed DCE packages.
  - b. You may see a requisite failure message stating that you cannot install the bos.rte.install 4.3.0.4 fileset. You can ignore the message and continue with the installation.
6. Exit SMIT.
  7. Log off as root.

**Note:** After installing the DCE Server software, you need to install the DCE PTF. This PTF is provided on the Supplemental Programs compact disc. For installation details, see “Installing the Fixes for the Prerequisite Products” on page 76.

## Configuring DCE Server

To configure the DCE Server:

1. Log in as root (if not already logged in).
2. Run the `mkdce` command to configure the DCE cell with Security and CDS servers. From a command prompt, enter:

```
/usr/lpp/dce/bin/mkdce -n cellname sec_srv cds_srv dts_global
```

Where *cellname* is the name of your local one-machine cell. For *cellname*, use “*hostname\_cell*”; for example, `myhost_cell`.

**Note:** During configuration, you are asked for the default password for DCE accounts. Do not use `cell_admin` as the password. Remember this password.

As `mkdce` runs, output is displayed. This output indicates that the configuration was successful with messages similar to:

```
Component Summary for Host: cbtest2.austin.ibm.com
Component      Configuration State  Running State
Security Master server  Configured           Running
Security client        Configured           Running
RPC                  Configured           Running
Initial Directory server  Configured           Running
Directory client       Configured           Running
DTS Global server      Configured           Running
```

```
The component summary is complete.
Configuration of DCE Host, cbtest.austin.ibm.com, was successful.
```

```
Configuration completed successfully.
```

3. At a shell prompt, enter:  

```
smitty dce
```

The DCE (Distributed Computing Environment) window is displayed.
4. Select **Configure DCE/DFS** and press the **Enter** key. The Configure DCE/DFS window is displayed.
5. Select **DCE/DFS Start Options** and press the **Enter** key. The DCE/DFS Start options window is displayed.
6. In the DCE/DFS Start options window:
  - a. Set **Start DCE now** to yes.
  - b. Set **Start DCE at system restart** to yes.
  - c. Press the **Enter** key.

## Installing and Configuring the DCE Client

This section contains information on installing and configuring the DCE Client. If your Component Broker server or client is to be installed on a computer that does not have a DCE server installed, you need to install the DCE client. Typically, the Component Broker client is installed on a DCE client system.

## Installing the DCE Client

**Note:** Before installing the DCE client, non-English users should untar the `dcd_dup_msg.tar` file. The tar file (located on the Supplemental Programs compact disc) contains additional code pages for non-English countries and includes the necessary message files as part of the installation images.

To install the DCE Client:

1. Log in as root (if not already logged in).
2. Insert the DCE compact disc into the CD-ROM drive.

**Note:** The DCE client can also be installed from the Supplemental Programs compact disc, from the `aix/dce2.2cli` directory.

3. At a shell prompt, enter:

```
smitty install_latest
```

The Install and Update from LATEST Available Software screen is displayed.

4. On this screen:
  - a. In the **INPUT device / directory for software** field, press the **F4** key to display a list of available devices of which you can select the appropriate input device (`/dev/cd0`) and press the **Enter** key. This screen is redisplayed with additional fields.
  - b. In the **SOFTWARE to install** field, press the **F4** key to display a list of the software packages contained on the compact disc.
  - c. In the SOFTWARE to install list, scroll down and select:
    - `dce.client`
    - `dce.compat`
    - `dce.pthreads`
  - d. Press the **Enter** key. The Install and Update from LATEST Available Software screen is redisplayed, but the software packages selected are in the **SOFTWARE to install** field.
  - e. Ensure that the **AUTOMATICALLY install requisite software?** and **EXTEND file systems if space needed?** fields are yes.
  - f. Press the **Enter** key. The ARE YOU SURE? screen is displayed.
5. On the ARE YOU SURE? screen, press the **Enter** key to continue.

### Notes:

- a. If any prerequisite failure messages are displayed, install the prerequisite software and reinstall the failed DCE packages.
- b. You may see a requisite failure message stating that you cannot install the `bos.rte.install 4.3.0.4` fileset. You can ignore the message and continue with the installation.

**Note:** After installing the DCE client software, you need to install the DCE PTF. This PTF is provided on the Supplemental Programs compact disc. For installation details, see “Installing the Fixes for the Prerequisite Products” on page 76.

## Configuring the DCE Client

Typically, many clients are configured to a single DCE cell. Configuring clients entails two distinct sets of operations. These tasks are separated into a split configuration of clients because a DCE cell administrator is unlikely to have root user access to every workstation in a cell. If you are both the cell administrator and the root user, you can combine these procedures as a single task, see “Full Configuration” on page 75.

If you are not the cell administrator and the root user, configure the DCE client using the split configuration. The two distinct sets of operations are:

- One required with cell administrator authority within the DCE cell. The cell administrator runs the administration portion from any machine in the cell to update the DCE CDS name space and security registry.
- One required with root user authority on the machine being configure. The root user of the DCE client machine runs the local portion to create the necessary files and to start the client daemons for all client components.

**Split Configuration:** Use the following distinct set of procedures to perform a split configuration. The split configuration is separated into the cell administrator task and root user task.

- Cell Administrator Task

This task must be performed by the DCE cell administrator, but it can be done from any machine in the cell.

1. At a shell prompt, enter:

```
smitty mkdceclient
```

The Type of configuration screen is displayed.

2. On this screen, select **3 admin only configuration for another machine**. The Administrator DCE Client Configuration screen is displayed.

3. On this screen:

- a. In the **CLIENTS to configure** field, select `sec_c1 cds_c1 dts_c1` from the list.
- b. In the **Cell ADMINISTRATOR'S account** field, type the account for the cell administrator if different from `cell_admin`.
- c. In the **Client Machine DCE HOSTNAME** field, type the local hostname as the DCE hostname for the client machine. If no name is specified, the fully-qualified TCP/IP hostname (including domain name) is used.
- d. In the **Client Machine IDENTIFIER** field, type the TCP/IP hostname or IP address for the client being configured.
- e. In the **LAN PROFILE** field, type the LAN profile that the client machine should use. Use this field only if the cell contains multiple LANs and requires the use of global DTS servers.
- f. Press the **Enter** key.

- After the cell administrator task completes, perform the root user task.

**Root User Task:** This task must be performed on the DCE client as root.

1. At a shell prompt, enter:

```
smitty mkdceclient
```

The Type of configuration screen is displayed.

2. On this screen, select **2 local only configuration for this machine**. The Local DCE/DFS Client Configuration screen is displayed.

3. On this screen:

- a. In the **CELL name** field, type the cell name of the Master Security Server for the cell.
- b. In the **CLIENTS to configure** field, select `rpc sec_c1 cds_c1 dts_c1` from the list.

- c. In the **SECURITY Server** field, type the TCP/IP hostname or IP address of the Master Security Server for the cell.
- d. In the **Client Machine DCE HOSTNAME** field, type the DCE hostname for the client machine. If no name is specified, the fully-qualified TCP/IP hostname (including domain name) is used.
- e. Press the **Enter** key.

**Full Configuration:** If you are both the cell administrator and the root user, you can perform the admin only and local only configuration at the same time. To perform a full configuration:

1. At a shell prompt, enter:

```
smitty mkdceclient
```

The Type of configuration screen is displayed.

2. On this screen, select **1 full configuration for this machine**. Full DCE/DFS Client Configuration screen is displayed.
3. On this screen:
  - a. In the **CELL name** field, type the cell name of the Master Security Server for the cell.
  - b. In the **CLIENTS to configure** field, select `rpc sec_c1 cds_c1 dts_c1` from the list.
  - c. In the **Client Machine DCE HOSTNAME** field, type the DCE hostname for the client machine. If no name is specified, the fully-qualified TCP/IP hostname (including domain name) is used.
  - d. In the **SECURITY Server** field, type the TCP/IP hostname or IP address of the Master Security Server for the cell.
  - e. In the **Cell ADMINISTRATOR'S account** field, type the account for the cell administrator if different from `cell_admin`.
  - f. In the **LAN PROFILE** field, type the LAN profile that the client machine should use. Use this field only if the cell contains multiple LANs and requires the use of global DTS servers.
  - g. Press the **Enter** key.

4. At a shell prompt, enter:

```
smitty dce
```

The DCE (Distributed Computing Environment) window is displayed.

5. Select **Configure DCE/DFS** and press the **Enter** key. The Configure DCE/DFS window is displayed.
6. Select **DCE/DFS Start Options** and press the **Enter** key. The DCE/DFS Start options window is displayed.
7. In the DCE/DFS Start options window:
  - a. Set **Start DCE now** to yes.
  - b. Set **Start DCE at system restart** to yes.
  - c. Press the **Enter** key.

**Additional Information:** When performing a `cell_admin` login on a new AIX DCE 2.2 installation, on either the server or the client side, after entering your password, you may see the following message:

```
Password must be changed!
```

```
DCE LOGIN SUCCESSFUL
```

```
Warning: This account has been marked by an administrator, recommending that the password be changed.
```

```
Do you wish to change now [y/n]? (y)
```

If you specify *y* (Yes, to change your password now), the system assists you to change your login password. If you specify *n* (No, to change your password later), each time you log on until your password is changed, the Password must be changed message displays.

To prevent the message from appearing without changing your password, perform the following steps:

1. At a command prompt, open the dcecp control program shell by entering:

```
dcecp
```

2. At the dcecp command prompt, enter:

```
account modify cell_admin -pwdvalid yes
```

3. End the DCE command session. Enter:

```
exit
```

4. Login to DCE again to verify that the message does not appear.

---

## Installing C Set++

C Set++ is required only for development on a Component Broker server or a C++ client. C Set++ can be found on the Software Development Solutions for AIX Version 4 compact disc that was shipped as part of AIX Version 4.2.1.

Use SMIT to install C Set++. The package to install is *x1C.C++*.

After installing C Set++, install the C Set++ PTF. This PTF is provided on the Supplemental Programs compact disc. For installation details, see "Installing the Fixes for the Prerequisite Products."

---

## Installing the Fixes for the Prerequisite Products

This section includes procedures for creating a CD-ROM file system and mounting this file system to install the PTFs. This step is required to install necessary fixes that are supplied on the Supplemental Programs compact disc. If you have already created a CD-ROM file system, skip the Creating a CD-ROM File System section.

The AIX PTFs are provided on the Supplemental Programs compact disc. The instructions contained in the Installing the PTFs section will update the prerequisite software installed on your system. If the initial software is not already installed on your system, the PTF for that software will not be installed. The PTFs provided on the Supplemental Programs compact disc are the following:

- Multi-Processor
- DCE Server and Client
- C Set++
- DB2

## Creating a CD-ROM File System

If a CD-ROM file system does not exist, use the following procedure to create one.

1. Log in as root (if not already logged in).

2. From a shell prompt, enter:

```
smitty crcdrfs
```

This displays the Add a CDROM File System screen.

3. On this screen:
  - a. In the **DEVICE name** field, press the **F4** key to lists the available devices. Select the appropriate device; for example:  
`cd0`
  - b. In the **MOUNT POINT** field, type the mount point; for example:  
`/cdrom`
  - c. In the **Mount AUTOMATICALLY at system restart?** field, press the **Tab** key to toggle from yes to no.
  - d. Press the **Enter** key.

## Installing the PTFs

**Note:** The perfagent bff only needs to be applied if the Performance Analysis and Control Commands were previously installed.

To install the PTFs, perform the following procedure:

1. Log in as root (if not already logged in).
2. Insert the Supplemental Programs compact disc into the CD-ROM drive.
3. Mount the file system. To mount, enter:  
`mount /cdrom`
4. From a shell prompt, enter:  
`smitty update_all`  
The Update Installed Software to Latest Level (Update All) screen is displayed.
5. On this screen:
  - a. In the **INPUT device / directory for software** field, type `/cdrom/aix/fixes`.
  - b. Press the **Enter** key to continue.
6. Press the **Enter** key, which displays the ARE YOU SURE? screen is displayed.
7. Press the **Enter** key to commit the install.
8. Press the **F10** key to exit SMIT.
9. Unmount the CD-ROM to eject the compact disc from the drive. To unmount, enter:  
`umount /cdrom`  
where `/cdrom` is the file system.
10. If you previously had DB2 databases on your computer, you must follow the steps in `/cdrom/aix/fixes/fixpack.readme` on the Supplemental Programs compact disc to migrate your databases to this new version of DB2. This is not required if this is a new installation of DB2.
11. Once the PTFs are installed, shut down the system and reboot. To reboot, enter:  
`shutdown -Fr`





---

## Chapter 6. Component Broker Migration

This section includes the following topics:

- “Migrating Object Builder Models”
- “Interface Changes That May Require Code Modifications”
- “Release-to-Release Compatibility” on page 80
- “Release-to-Release Interoperability” on page 80
- “Migration Considerations for OS/390 Servers” on page 80

---

### Migrating Object Builder Models

Model information developed using prior releases of the Object Builder tool will be automatically migrated to this release when opened. When you save the migrated model, it will be saved in the new Release 2.0 format. This format is not supported by earlier releases of the Object Builder tool. Make a backup of the model if you may need to use the model on a Release 1.3 system.

**Warning:** The model release information is maintained in .uni files. If these .uni files are stored in a subdirectory in the Component Broker directory tree, they will be deleted when Component Broker is uninstalled.

---

### Interface Changes That May Require Code Modifications

The following interfaces could require changes:

#### **getConfigInfo() usage**

The IBOIMExtLocalToServer::IMSpecializedHomeDataObject interface is not available in release 2.0. It was replaced with the ManagedAdvancedServer::ISpecializedHomeDataObject interface. This change affects programs that were using the getConfigInfo() method for accessing the userData field in the definition of the specialized home. If you were previously narrowing to the old interface as follows:

```
yD0 = IBOIMExtLocalToServer::IMSpecializedHomeDataObject::_narrow(theD0);
```

You will have to change your code as follows:

```
myD0 = IManagedAdvancedServer::ISpecializedHomeDataObject::_narrow(theD0)
```

#### **nbind() and unbind\_with\_string() of naming contexts**

In Release 2.0, the unbind() and unbind\_with\_string() methods do not allow the unbind if it was bound using bind\_new\_context because this would result in an orphaned name context. If your program uses the unbind() or unbind\_with\_string() methods, you may need to modify it to handle the resulting CORBA::PERSIST\_STORE exceptions. For more information, see the “NamingContext:rebind()” section in the *Component Broker Programming Reference*.

Because of the addition of the #pragma omg.org line to IDL, many of the symbols generated by IDL2COM could change. This is because IDL2COM will generate properly qualified names based on the #pragma value. Client applications may need to be changed to properly match the new values. For example, prior to this change, the VersionIndependentProgID for CosNaming.Binding might have looked like the following example in the Windows registry:

```
IDL:CosNaming.Binding
```

After the change it might look like the following:

```
IDL:omg.org.CosNaming.Binding
```

If you have Visual Basic or C++ source code that uses values similar to the examples, you will have to change to add the new omg.org portion. These changes refer to any of the COM Wrappers supplied by IBM, and to any others you may have generated based on IBM's supplied IDL.

---

## Release-to-Release Compatibility

The server portion of Component Broker Release 1.3 models must be migrated, regenerated, and recompiled using Component Broker Release 2.0. Binaries produced using Release 1.3 or earlier will not run on Release 2.0 without recompilation or regeneration.

**Note:** Refer to the *Component Broker for Windows NT and AIX CICS and IMS Application Adaptor Quick Beginnings* for CICS and IMS application adaptor model migration issues.

---

## Release-to-Release Interoperability

Component Broker Release 2.0 servers and clients do not interoperate with Component Broker Release 1.3 or earlier clients and servers. For Release 2.0, you must migrate the clients and the servers to which they connect at the same time.

---

## Migration Considerations for OS/390 Servers

If you plan to migrate your application to the OS/390 Component Broker server in the future, please note the following:

1. UUIDs created on the workstation clients are not usable to create or find OS/390 Component Broker server transient Business Objects. If you want to use UUIDs, use specialized homes to generate them so that the UUIDs are created on the same platform as the server. Do not create the UUIDs on the client.
2. Nil object references in C++ are implemented slightly differently between the platforms. But if you follow the programming model of initializing with `Object::_nil` and testing for a nil value by using the `CORBA::is_nil()` method (as opposed to using the value of `NULL`), you will not notice the difference.

---

## Chapter 7. Performing an Unattended Installation on Windows NT

You can install Component Broker without the need to monitor the installation and without the requirement to provide input to the installation dialog. The procedure uses response files to provide the required information to the installation program.

Component Broker includes sample response files (each response file is named `setup.iss`) for each of the following install images:

Image	Location
Server	In the root directory on the package CD.
Client	In the platform-specific directory (WinNT or Win95) on the package CD.
Toolkit	In the root directory on the package CD.
CICS and IMS Application Adaptor	In the Windows NT platform-specific directory on the package CD.
Oracle Application Adaptor	In the Windows NT platform-specific directory on the package CD.

For the Server compact disc, the sample response file is configured for a System Manager installation. To change the installation option for other types of installation, instructions are included in the sample response file.

For the Client compact disc, the sample response file is configured for a Typical Client. To change the installation option for another type of installation, instructions are included in the sample response file.

For the Toolkit compact disc, the sample response file is configured to install all the packages on the compact disc except the CICS and IMS Application Adaptor SDK. To install one or more specific packages, refer to the instructions included in the sample response file.

For the CICS and IMS Application Adaptor and the Oracle Application Adaptor compact discs, the sample response file is configured to install the run-time package.

---

### Preparing Your Response File

Perform the following steps to prepare your response file:

1. Copy the response file to your hard disk. For example, to copy the response file for installing a server on a Windows NT computer, enter:

```
copy x:\setup.iss y:\target\server.iss
```

Where *x*: represents the designation for your CD-ROM drive, *y*: represents your hard disk, *target* represents the target directory on your hard disk, and *server* represents the name you assign to the copied response file.

2. Edit the copied response file (*server.iss*) and update the information for your installation selection. Your response file must meet the following requirements:
  - The file must be in the same format as the sample response file.
  - The dialogs must be listed in the correct order.
  - All required information must be included correctly in the response file.

The response file includes the following sections:

- The header section
- The dialog sequence section
- The dialog data section

Do not make any changes to the header section. You should start with the dialog data section, then return to the dialog sequence section.

In the dialog data section, there are one or more dialog sections. Each dialog section corresponds to one window in the installation program. Each dialog section starts with a unique dialog name enclosed with a square bracket, for example:

```
[SdWelcome-0]
```

The dialog name is made up of the type of dialog, followed by a hyphen (-) and the next sequential order for this type of dialog in this installation. The types of dialog used by Component Broker are:

<b>SdWelcome</b>	Welcome dialog
<b>AskDestPath</b>	Destination directory dialog
<b>AskOptions</b>	Option dialog
<b>MessageBox</b>	Message dialog
<b>SdAskOptionsList</b>	Option dialog for custom install
<b>AskYesNo</b>	Question dialog with <b>Yes</b> and <b>No</b> buttons
<b>SdShowDlgEdit1</b>	Edit field (Input) dialog
<b>SdStartCopy</b>	Verify setup dialog
<b>SdFinish</b>	Reboot dialog

Depending on the type of dialog, the dialog data entry list for each type of dialog is different except for the `Result=1` data entry. The `Result=1` data entry represents the **Next** or **Finish** buttons and occurs at the end of each dialog section. Refer to the sample response file for the data entry items used by each type of dialog.

The following example illustrates four dialogs. For clarity, the dialog names are highlighted:

```
[SdWelcome-0]  
Result=1  
[AskDestPath-0]  
szPath=C:\BROKER  
Result=1  
[AskOptions-0]  
Result=1  
Sel-0=1  
Sel-1=0  
Sel-2=0  
Sel-3=0  
[SdShowDlgEdit1-0]  
szEdit1=C:\Program Files\IBM\GSK  
Result=1
```

The dialog sections must appear in the response file in the same order as their corresponding windows display during an attended install. If the dialogs appear in an incorrect order, a -12 error code is generated and is appended to the setup.log file.

You can make changes to the existing sample response file by commenting out, uncommenting, or adding new dialog sections. When you are finished working with the dialog sections in the dialog data section, return to the dialog sequence section and ensure that there is an entry for each dialog name used in the dialog section. The entry must appear in the following format:

```
Dlg#=dialog_name
```

Where # represents the dialog order number in this installation; *dialog\_name* represents the dialog name. The dialog order numbers start with 0 (zero) and dialog 0 is always the Welcome dialog (for example, Dlg0=SdWelcome-0). Each dialog after dialog 0 increments the order number by 1. Make sure the order of dialogs is the same as in an attended install. After putting in the the order of the dialogs, adjust the dialog count in the dialog sequence section to reflect the exact number of dialogs used in this installation.

The following is an example of the dialog sequence section for the dialog data sections in the previous example:

```
[DlgOrder]
Dlg0=SdWelcome-0
Dlg1=AskDestPath-0
Dlg2=AskOptions-0
Dlg3=SdShowDlgEdit1-0
Count=4
```

Additional instructions are included in the response file as comments. For a sample response file, see Appendix A, “Sample Response File” on page 169.

---

## Running the Unattended Installation

Run setup with the following arguments:

```
setup -s -fcompile_script_path
      -f1response_file_path
      -f2log_file_path
```

Where:

- *compile\_script\_path* represents the name and path where the Component Broker compile script (setup.ins) is located.
- *response\_file\_path* represents the name and path where your response file is located.
- *log\_file\_path* represents the name of the log file that is created when you run setup.exe and the path where you want the file to be created.

For example, you could run the unattended installation using commands similar to the following:

x:

```
setup -s -f1y:\target\server.iss
      -f2y:\target\setup.log
```

Where x: represents the CD-ROM drive, y:\target represents a directory on your hard drive, server.iss represents the edited response file, and setup.log represents the name of the log file.

**Notes:**

1. The -f parameter is optional if *compile\_script\_path* is located in the directory in which setup.exe is located (the default directory).
2. The -f1 parameter is optional if the response file is located in the default directory **and** if the response file is named setup.iss.
3. If the -f1 parameter is specified with the -f parameter, -f1 must follow -f.
4. The -f2 parameter is optional if you want the log file to be created in the default directory.
5. The -f2 parameter, if specified, must be the final parameter.

After running the unattended installation, the computer automatically reboots. If this is a first time installation for Component Broker, the CICS and IMS application adapter, or the Oracle application adaptor, the readme is displayed automatically after the computer reboots. In addition, the configuration tool is automatically started if your installation includes any of the following:

- Server
- C++ Client
- ActiveX Client
- System Manager
- System Management Agent
- System Manager User Interface
- CICS and IMS application adapter
- Oracle application adaptor

---

## Using the Log File

A log file (containing information about the result of the installation) is created each time you perform an unattended installation. The name of the log file defaults to setup.log and is located in the default directory unless you specify a different name and location using the -f2 parameter.

You can verify the success of your unattended installation by checking the **ResultCode** value in the **[ResponseResult]** section of the log file. The **ResultCode** can be one of the following:

Result Code	Description
0	The installation was successful.
-1	A general error occurred.
-3	Required data was not found in the response file.
-4	There was insufficient memory available for the installation.
-5	The response file does not exist.
-9	A list type (string or number) is not valid.
-10	A data type is not valid.
-11	There was an unknown error during the setup procedure.
-12	The dialogs are not in the correct order.

---

## Chapter 8. Installing Component Broker for Windows NT

### Note

Before installing Component Broker, create an Emergency Repair Disk (ERD) using the RDISK.EXE utility provided with the operating system.

This chapter contains the following procedures for installing Component Broker packaged components and prerequisite software:

- “Installing Component Broker”
- “Installing the Development Environment” on page 108
- “Applying the Component Broker PTF” on page 110
- “Accessing the Online Documentation” on page 112
- “Adding Components to an Existing Installation” on page 112
- “Reconfiguring Your Component Broker Installation” on page 113
- “Uninstalling Component Broker” on page 117
- “Uninstall Failure Cleanup” on page 117

### Notes:

1. Before installing Component Broker, uninstall any pre-existing versions.
2. See the “Installation and Configuration” section of the *Late Breaking News* provided with Component Broker for additional important setup information.

If installing Component Broker on multiple computers, consider:

- Installing the System Manager as your first installation. This computer is to be both a Component Broker server, a System Manager host, and a bootstrap host to identify the naming servers. This computer manages other Component Broker clients and hosts.
- Performing the Application Server installation on other computers that you want to be Component Broker servers. During the installation, you must provide the System Manager port ID, the host name, and information about a bootstrap host. Use the information you specified during the System Manager procedure.
- Increasing the time-out value of the ORB to 180 seconds after you install Component Broker to avoid time-outs while running long-running methods. You can also set the time-out value to 0 to define unlimited time access. For information on changing the time-out value, see the Component Broker online documentation.

---

## Installing Component Broker

This section discusses the available installation options for installing the Component Broker run time, several pre-installation considerations, and includes the following run-time installation scenarios:

- Standalone Workstation
- System Manager
- Application Server
- Custom Install

## Installation Options

This section details the available options for installing Component Broker. The following list contains each of the Component Broker options with the software it installs.

- System Manager
  - Component Broker Server
  - C++ Client
  - System Manager
  - System Manager User Interface
  - DB2 Application Adaptor
  - Documentation
- Standalone Workstation
  - Component Broker Server
  - C++ Client
  - System Manager
  - System Manager User Interface
  - DB2 Application Adaptor
  - Documentation
- Application Server
  - Component Broker Server
  - C++ Client
  - System Management Agent
  - DB2 Application Adaptor
- Custom Install
  - Any valid combination of software

## Pre-Installation

Before performing any Component Broker installation:

1. Read Chapter 3, “Planning for the Component Broker Installation” on page 23.
2. Read Chapter 4, “Installing the Prerequisite Software on Windows NT” on page 39 and install all the Component Broker prerequisite products.
3. Verify that you have administrator authority under Windows NT. Your installation creates a new NT Service (CBConnector) that requires Administrator authority.
4. Disable any antiviral programs, unless you are using IBM AntiVirus 3.0. Failure to do so could result in an incomplete installation. After a Component Broker installation, you can reactivate your antiviral programs.
5. To install Component Broker on a system with a different TCP/IP domain than that of the system on which the Component Broker System Manager is installed, you must add the new domain to the list of domains for which to search in the TCP/IP configuration on the System Manager's system.
6. A client install does not require DB2 client or server.

Any installation scenario that installs the Component Broker server detects to see if JDK 1.1.6 or above was previously installed on the system. If not, the installation will terminate, and you must install JDK 1.1.6 before continuing the Component Broker installation. If JDK 1.1.6 or above was previously installed, you need to ensure that the classpath and path include the directories where the JDK is installed. Perform the following steps:

1. From the **Start** menu, select **Settings** → **Control Panel**
2. Open the System object.



3. In the System Properties window, select the **Environment** tab.
4. Verify that the User Variables includes the following in the path statement:

`x:\JDK1.1.6\bin;`

and includes the following in the classpath statement:

`x:\JDK1.1.6\lib\classes.zip;`

Where *x* represents the disk on which the JDK is installed and JDK1.1.6 is the JDK base directory. If the variables do not include the paths, update the variables.

The Component Broker Configuration tool fails if these environment variables are not set.

## Installing the System Manager Option

Perform the following steps to install the System Manager option:

1. Change to the desired locale before continuing. Once the server is installed, you should not switch to a different locale unless you plan to uninstall the server.
2. Uninstall any previous version of Component Broker.
3. Insert the CBBConnector compact disc into your CD-ROM drive.
4. Display the contents of the compact disc.
5. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To install Component Broker, you must click the Setup icon that looks like a computer. The Component Broker for Windows NT splash screen is displayed followed by the Welcome window.
6. On the Welcome window, click the **Next** button. A message is displayed indicating that Setup is searching for Component Broker packages on your system, then the Choose Destination Location window is displayed.
7. In the Choose Destination Location window:
  - a. The default installation directory is CBroker. To change the drive or directory, click the **Browse** button to open the Choose Directory window in which you can specify a different location.
  - b. In the Choose Directory window:
    - 1) Type a directory name.
    - 2) Click the **OK** button to continue.
    - 3) If the specified directory does not exist, Setup displays a window asking if you want the directory to be created. Click the **Yes** button.
  - c. The Choose Destination Location window is displayed again and lists the directory you specified. Click the **Next** button to continue.
8. In the Select Installation Type window:
  - a. Select the System Manager installation option.
  - b. Click the **Next** button to continue.

**Notes:**

- a. The installation program verifies that your selections are compatible. If they are not compatible, the program displays a message that identifies the conflict.
  - b. If DCE 2.0 is not installed, the install program asks if you want to install it. DCE is a prerequisite for Component Broker.
    - Click the **Yes** button to stop the Component Broker install. After the installation stops, you can install DCE. After installing DCE, restart the Component Broker for Windows NT installation.
    - Click the **No** button to choose another installation selection.
  - c. If you install the ActiveX client or the Java client for development, remember to install the ActiveX or Java client SDK respectively.
  - d. If you install the ActiveX client with the C++ client, you need to create a two-compiler environment; for details, see Appendix D, “The Two-Compiler Environment (NT only)” on page 189.
9. In the Enter SSL Runtime Install Path window:
- a. The default installation directory for the SSL run time is GSK. To change the drive or directory, specify a different directory in the **Install Path** field.

**Note:** Avoid installing the SSL run time under the Component Broker directory. If the SSL run time is also registered to be used by other products, the cleanup of the Component Broker directory will fail.
  - b. Click the **Next** button to continue.
- Note:** This window does not display if the SSL run time has been installed previously on the system.
10. In the Enter NetQuestion Install Path window:
- a. The default installation directory for NetQuestion is IMNNQ. To change the drive or directory, specify a different directory in the **Install Path** field.
  - b. Click the **Next** button to continue.
- Note:** This window does not display if NetQuestion has previously been installed on the system. For example, if you have the DB2 V.5 documentation installed, NetQuestion is found in the `x:\imnnq_nt` directory (where `x:` is your DB2 installation drive). The Component Broker installation copies additional files to this location.
11. In the Select Application Adaptor window:
- a. Verify that the DB2 check box is set.
  - b. Click the **Next** button to continue.
12. In the Verify Configuration Setting window, verify that the items to be installed are correct.
- Click the **Back** button to review or change any information you previously entered.
  - Click the **Next** button to start the configuration.
13. A status bar is displayed indicating the progress of each component file transfer. The installation may take several minutes.
14. A window is displayed stating that setup is installing the SSL run time. When the SSL run time installation has completed, the window closes automatically.
15. A window is displayed stating that the setup program is installing the NetQuestion search engine. When the NetQuestion installation has completed, the window closes automatically.
16. A window displays stating that the Component Broker installation has completed. When you exit this window, your computer automatically restarts. Optionally in this window you can set check boxes to:

- Automatically launch the configuration tool when the computer restarts
- Automatically display the readme file when the computer restarts

Click the **Finish** button to restart your computer.

## Configuring the System Manager Option

Before beginning the configuration, verify the following:

- The JDK 1.1.6 classes.zip is included in the CLASSPATH environment variable before any previous versions of the JDK.
- The JDK 1.1.6 bin directory is in the PATH environment variable.

Perform the following steps to configure your System Manager installation:

1. If your computer starts the configuration tool automatically when you restart your computer after performing the installation, skip to the next step. To start the configuration tool from the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Configuration tool**. The Configure New Install window opens.
2. In the Configure New Install window:
  - a. Select the **New Install Configuration** radio button to configure your entire Component Broker installation.
  - b. Select the **Next** button to continue.

### 3. If the DB2 client is installed

In the Enter Remote DB2 Server Information window:

- a. Type the DB2 user ID used to access the remote DB2 server in the **User ID** field.
  - b. Type the password for the DB2 user ID in the **Password** field.
  - c. Specify the fully-qualified host name of the server in the **Server Host** field. For example:  
`db2srvr1.austin.ibm.com`
  - d. Type the node for this computer in the **Local Node** field.
  - e. Click the **Next** button to continue.
4. In the Enter Client's Bootstrap Host Name and Port Number window:
    - a. Type the fully-qualified client bootstrap **Host Name** (for example, myhost.austin.ibm.com). The default host name is the local host.  
**Note:** The bootstrap host computer contains information about the name servers. Any Component Broker server can act as a bootstrap host.
    - b. The default **Port ID** is 900. Accept the default or type a different port ID.
    - c. Click the **Next** button to continue.
  5. In the Enter Component Broker Secure Socket Layer (SSL) Port ID window:
    - a. The default port ID is 3004. This is the port ID that Component Broker will use to perform SSL security. Accept the default or supply a different port ID.
    - b. Click the **Next** button to continue.
  6. In the Input User ID And Password For Component Broker Service window:
    - a. Select a user ID from the list or type a user ID for starting the Component Broker service in the **User ID** field.
    - b. Type the password associated with the user ID in the **Password** field.

- c. Click the **Next** button to continue.

**Notes:**

- a. If the selected user ID was created by the Administrator, the password entered must be identical to the existing password. If the password entered does not match the existing user ID, System Management configuration will fail. If the existing user does not have the advanced user right of “logon as a service,” the installation program grants this authority automatically.

- b. When you uninstall, this user ID and password are not deleted. If you reinstall, you must specify this user ID and password or a new user ID and password combination. If the password entered does not match an existing user ID, System Management configuration will fail.

7. In the Enter DCE Cell Administrator User ID and Password window:

- a. Type the DCE cell administrator ID in the **User ID** field. The default ID is `cell_admin`.

- b. Type the DCE administrator password in the **Password** field.

- c. Click the **Next** button to continue.

8. In the Enter System Management Host Name and Port Number window:

- a. Type the fully-qualified host name of the computer on which system manager is installed. The default is your local host.

- b. Type a number representing the port the System Manager User Interface will use to communicate with other System Managers and System Management Agents on all the connected computers. The default port ID is 20002. You can accept the default or supply a different port ID.

- c. Click the **Next** button to continue.

9. In the Verify Configuration Setting window, verify that the items to be installed are correct.

- Click the **Back** button to review or change any information you previously entered.
- Click the **Next** button to start the configuration.

10. The Configuration Status window displays information about the items being configured. The configuration process creates the `cbconfig.log` file in the `CBroker\service` directory (where *CBroker* represents the directory in which Component Broker is installed) and places all status information into this log file.

**Notes:**

- a. If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.

- b. If System Management configuration fails with the following error:

```
SM - Failed to start CBConnector service. ReturnCode = 1069
```

Refer to “Return Code 1069 From System Manager Configuration” on page 142 for information about fixing the problem. After the problem is fixed, you can reconfigure your Component Broker installation.

11. An informational window is displayed stating that the system management configuration completed successfully. Click the **OK** button to continue.

12. **If the DB2 server is installed on this computer**

As part of the configuration process, several windows are opened that show the configuration of DB2 for the Component Broker server:

- After the system management configuration has completed, a status window is displayed stating that Setup has finished launching the DB2 configuration process. Click the **OK** button to continue.
- After the configuration, two DB2 windows remain open (the other windows close themselves).

If these windows show successful messages at the end, close these windows.

If either window contains an error message, double check the following areas:

- Ensure that the correct transaction monitor is configured by opening a DB2 command line processor window. Enter:

```
get dbm cfg
```

and verifying that the transaction processor monitor name is `somtrx1i`. If not, enter:

```
update database manager configuration using TP_MON_NAME somtrx1i
```

- To verify that the CBSRFCDB has been created, enter:

```
list db directory
```

If the CBSRFCDB has not been created, enter:

```
quit  
db2 -f x:\CBroker\etc\inscrtdb.db2
```

where `x:\CBroker` is the base directory where Component Broker is installed.

The DB2 configuration accomplishes:

- The set up of the Database Manager configuration `TP_MON_NAME` to `somtrx1i`, the Component Broker transaction monitor.
- The creation of a database CBSRFCDB for Component Broker Reference Collections database.

13. An informational window is displayed stating that the configuration has completed. Click the **OK** button to close the window.

**Note:** If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.

14. Exit the Component Broker Configuration tool. In the Configuration Status window, click either the **Close** button or the **Cancel** button.

15. If you are using a non-English locale, you must configure the Interface Repository for the current user. The pre-populated Interface Repository (IR) database that is installed on a server is compatible only with the English (United States) setting in the Regional Setting selection of the NT Control Panel. If any other regional setting is selected, you must rebuild the IR using that particular regional setting. To rebuild the IR, perform the following steps:

- a. If not in the desired locale, change to that locale now.

- b. Copy the files. From a command prompt, enter:

```
copy \CBroker\etc\irempty \CBroker\etc\irwork
```

- c. Rebuild the IR using the new files. From a command prompt, enter:

```
irpop1
```

On completion, enter:

```
irpop2
```

This procedure could take from ten to fifteen minutes to complete. If multiple servers are needed and the populate time is a consideration, the database files from `irwork` may be copied between servers. This should be done before applications are installed on either the source or target server, because installing an application could modify the IR.

Accessing the IR under any setting other than English (United States) without this correction may generate a `SOMDERROR_IRIncoherent` error.

## Installing the Standalone Workstation

Perform the following steps to install the Standalone Workstation option:

1. Change to the desired locale before continuing. Once the server is installed, you should not switch to a different locale unless you plan to uninstall the server.
2. Uninstall any previous version of Component Broker.
3. Insert the CBConnector compact disc into your CD-ROM drive.
4. Display the contents of the compact disc.
5. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To install Component Broker, you must click the Setup icon that looks like a computer. The Component Broker for Windows NT splash screen is displayed followed by the Welcome window.
6. On the Welcome window, click the **Next** button. A message is displayed indicating that Setup is searching for Component Broker packages on your system, then the Choose Destination Location window is displayed.
7. In the Choose Destination Location window:
  - a. The default installation directory is CBroker. To change the drive or directory, click the **Browse** button to open the Choose Directory window in which you can specify a different location.
  - b. In the Choose Directory window:
    - 1) Type a directory name.
    - 2) Click the **OK** button to continue.
    - 3) If the specified directory does not exist, Setup displays a window asking if you want the directory to be created. Click the **Yes** button.
  - c. The Choose Destination Location window is displayed again and lists the directory you specified. Click the **Next** button to continue.
8. In the Select Installation Type window:
  - a. Select the Standalone Workstation installation option.
  - b. Click the **Next** button to continue.

### Notes:

- a. The installation program verifies that your selections are compatible. If they are not compatible, the program displays a message that identifies the conflict.
  - b. If DCE 2.0 is not installed, the install program asks if you want to install it. DCE is a prerequisite for Component Broker.
    - Click the **Yes** button to stop the Component Broker install. After the installation stops, you can install DCE. After installing DCE, restart the Component Broker for Windows NT installation.
    - Click the **No** button to choose another installation selection.
9. In the Enter SSL Runtime Install Path window:
    - a. The default installation directory for the SSL run time is GSK. To change the drive or directory, specify a different directory in the **Install Path** field.

**Note:** Avoid installing the SSL run time under the Component Broker directory. If the SSL run time is also registered to be used by other products, the cleanup of the Component Broker directory will fail.
    - b. Click the **Next** button to continue.

**Note:** This window does not display if the SSL run time has been installed previously on the system.

10. In the Enter NetQuestion Install Path window:

- a. The default installation directory for NetQuestion is IMNNQ. To change the drive or directory, specify a different directory in the **Install Path** field.
- b. Click the **Next** button to continue.

**Note:** This window does not display if NetQuestion has previously been installed on the system. For example, if you have the DB2 V.5 documentation installed, NetQuestion is found in the `x:\imnnq_nt` directory (where `x`: is your DB2 installation drive). The Component Broker installation copies additional files to this location.

11. In the Select Application Adaptor window:

- a. Verify that the DB2 check box is set.
- b. Click the **Next** button to continue.

12. In the Verify Configuration Setting window, verify that the items to be installed are correct.

- Click the **Back** button to review or change any information you previously entered.
- Click the **Next** button to start the configuration.

13. A status bar is displayed indicating the progress of each component file transfer. The installation may take several minutes.

14. A window is displayed stating that setup is installing the SSL run time. When the SSL run time installation has completed, the window closes automatically.

15. A window is displayed stating that the setup program is installing the NetQuestion search engine. When the NetQuestion installation has completed, the window closes automatically.

16. A window displays stating that the Component Broker installation has completed. When you exit this window, your computer automatically restarts. Optionally in this window you can set check boxes to:

- Automatically launch the configuration tool when the computer restarts
- Automatically display the readme file when the computer restarts

Click the **Finish** button to restart your computer.

## Configuring the Standalone Workstation

Before beginning the configuration, verify the following:

- The JDK 1.1.6 classes.zip is included in the CLASSPATH environment variable before any previous versions of the JDK.
- The JDK 1.1.6 bin directory is in the PATH environment variable.

Perform the following steps to configure your Standalone Workstation installation:

1. If your computer starts the configuration tool automatically when you restart your computer after performing the installation, skip to the next step. To start the configuration tool from the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Configuration tool**. The Configure New Install window opens.

2. In the Configure New Install window:

- a. Select the **New Install Configuration** radio button to configure your entire Component Broker installation.
- b. Select the **Next** button to continue.

3. **If the DB2 client is installed**

In the Enter Remote DB2 Server Information window:

- a. Type the DB2 user ID used to access the remote DB2 server in the **User ID** field.
- b. Type the password for the DB2 user ID in the **Password** field.
- c. Specify the fully-qualified host name of the server in the **Server Host** field. For example:  
`db2srvr1.austin.ibm.com`
- d. Type the node for this computer in the **Local Node** field.
- e. Click the **Next** button to continue.

4. In the Enter Client's Bootstrap Host Name and Port Number window:

- a. Type the fully-qualified client bootstrap **Host Name** (for example, `myhost.austin.ibm.com`). The default host name is the local host.

**Note:** The bootstrap host computer contains information about the name servers. Any Component Broker server can act as a bootstrap host.

- b. The default **Port ID** is 900. Accept the default or type a different port ID.
- c. Click the **Next** button to continue.

5. In the Enter Component Broker Secure Socket Layer (SSL) Port ID window:

- a. The default port ID is 3004. This is the port ID that Component Broker will use to perform SSL security. Accept the default or supply a different port ID.
- b. Click the **Next** button to continue.

6. In the Input User ID And Password For Component Broker Service window:

- a. Select a user ID from the list or type a user ID for starting the Component Broker service in the **User ID** field.
- b. Type the password associated with the user ID in the **Password** field.
- c. Click the **Next** button to continue.

**Notes:**

- a. If the selected user ID was created by the Administrator, the password entered must be identical to the existing password. If the password entered does not match the existing user ID, System Management configuration will fail. If the existing user does not have the advanced user right of "logon as a service," the installation program grants this authority automatically.
- b. When you uninstall, this user ID and password are not deleted. If you reinstall, you must specify this user ID and password or a new user ID and password combination. If the password entered does not match an existing user ID, System Management configuration will fail.

7. In the Enter DCE Cell Administrator User ID and Password window:

- a. Type the DCE cell administrator ID in the **User ID** field. The default ID is `cell_admin`.
- b. Type the DCE administrator password in the **Password** field.
- c. Click the **Next** button to continue.

8. In the Enter System Management Host Name and Port Number window:

- a. Type the fully-qualified host name of the computer on which system manager is installed. The default is your local host.
- b. Type a number representing the port the System Manager User Interface will use to communicate with other System Managers and System Management Agents on all the connected computers. The default port ID is 20002. You can accept the default or supply a different port ID.



- c. Click the **Next** button to continue.
9. In the Verify Configuration Setting window, verify that the items to be installed are correct.
    - Click the **Back** button to review or change any information you previously entered.
    - Click the **Next** button to start the configuration.
  10. The Configuration Status window displays information about the items being configured. The configuration process creates the cbconfig.log file in the *CBroker*\service directory (where *CBroker* represents the directory in which Component Broker is installed) and places all status information into this log file.

**Notes:**

- a. If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.
  - b. If System Management configuration fails with the following error:
 

```
SM - Failed to start CBConnector service. ReturnCode = 1069
```

 Refer to “Return Code 1069 From System Manager Configuration” on page 142 for information about fixing the problem. After the problem is fixed, you can reconfigure your Component Broker installation.
11. An informational window is displayed stating that the system management configuration completed successfully. Click the **OK** button to continue.

**12. If the DB2 server is installed on this computer**

As part of the configuration process, several windows are opened that show the configuration of DB2 for the Component Broker server:

- After the system management configuration has completed, a status window is displayed stating that Setup has finished launching the DB2 configuration process. Click the **OK** button to continue.
- After the configuration, two DB2 windows remain open (the other windows close themselves).

If these windows show successful messages at the end, close these windows.

If either window contains an error message, double check the following areas:

- Ensure that the correct transaction monitor is configured by opening a DB2 command line processor window. Enter:

```
get dbm cfg
```

and verifying that the transaction processor monitor name is somtrx1i. If not, enter:

```
update database manager configuration using TP_MON_NAME somtrx1i
```

- To verify that the CBSRFCDB has been created, enter:

```
list db directory
```

If the CBSRFCDB has not been created, enter:

```
quit
db2 -f x:\CBroker\etc\inscrtdb.db2
```

where x:\CBroker is the base directory where Component Broker is installed.

The DB2 configuration accomplishes:

- The set up of the Database Manager configuration TP\_MON\_NAME to somtrx1i, the Component Broker transaction monitor.
- The creation of a database CBSRFCDB for Component Broker Reference Collections database.

13. An informational window is displayed stating that the configuration has completed. Click the **OK** button to close the window.

**Note:** If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.

14. Exit the Component Broker Configuration tool. In the Configuration Status window, click either the **Close** button or the **Cancel** button.
15. If you are using a non-English locale, you must configure the Interface Repository for the current user. The pre-populated Interface Repository (IR) database that is installed on a server is compatible only with the English (United States) setting in the Regional Setting selection of the NT Control Panel. If any other regional setting is selected, you must rebuild the IR using that particular regional setting. To rebuild the IR, perform the following steps:

- a. If not in the desired locale, change to that locale now.

- b. Copy the files. From a command prompt, enter:

```
copy \CBroker\etc\irempty \CBroker\etc\irwork
```

- c. Rebuild the IR using the new files. From a command prompt, enter:

```
irpop1
```

On completion, enter:

```
irpop2
```

This procedure could take from ten to fifteen minutes to complete. If multiple servers are needed and the populate time is a consideration, the database files from irwork may be copied between servers. This should be done before applications are installed on either the source or target server, because installing an application could modify the IR.

Accessing the IR under any setting other than English (United States) without this correction may generate a SOMDERROR\_IRIncoherent error.

## Installing the Application Server Option

Perform the following steps to install the Application Server option:

1. Change to the desired locale before continuing. Once the server is installed, you should not switch to a different locale unless you plan to uninstall the server.
2. Uninstall any previous version of Component Broker.
3. Insert the CConnector compact disc into your CD-ROM drive.
4. Display the contents of the compact disc.
5. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To install Component Broker, you must click the Setup icon that looks like a computer. The Component Broker for Windows NT splash screen is displayed followed by the Welcome window.
6. On the Welcome window, click the **Next** button. A message is displayed indicating that Setup is searching for Component Broker packages on your system, then the Choose Destination Location window is displayed.
7. In the Choose Destination Location window:
  - a. The default installation directory is CBroker. To change the drive or directory, click the **Browse** button to open the Choose Directory window in which you can specify a different location.
  - b. In the Choose Directory window:
    - 1) Type a directory name.

- 2) Click the **OK** button to continue.
  - 3) If the specified directory does not exist, Setup displays a window asking if you want the directory to be created. Click the **Yes** button.
  - c. The Choose Destination Location window is displayed again and lists the directory you specified. Click the **Next** button to continue.
8. In the Select Installation Type window:
- a. Select the Application Server installation option.
  - b. Click the **Next** button to continue.

**Notes:**

- a. The installation program verifies that your selections are compatible. If they are not compatible, the program displays a message that identifies the conflict.
  - b. If DCE 2.0 is not installed, the install program asks if you want to install it. DCE is a prerequisite for Component Broker.
    - Click the **Yes** button to stop the Component Broker install. After the installation stops, you can install DCE. After installing DCE, restart the Component Broker for Windows NT installation.
    - Click the **No** button to choose another installation selection.
  - c. If you install the ActiveX client or the Java client for development, remember to install the ActiveX or Java client SDK respectively.
  - d. If you install the ActiveX client with the C++ client, you need to create a two-compiler environment; for details, see Appendix D, "The Two-Compiler Environment (NT only)" on page 189.
9. In the Enter SSL Runtime Install Path window:
- a. The default installation directory for the SSL run time is GSK. To change the drive or directory, specify a different directory in the **Install Path** field.

**Note:** Avoid installing the SSL run time under the Component Broker directory. If the SSL run time is also registered to be used by other products, the cleanup of the Component Broker directory will fail.
  - b. Click the **Next** button to continue.
- Note:** This window does not display if the SSL run time has been installed previously on the system.
10. In the Select Application Adaptor window:
- a. Verify that the DB2 check box is set.
  - b. Click the **Next** button to continue.
11. In the Verify Configuration Setting window, verify that the items to be installed are correct.
- Click the **Back** button to review or change any information you previously entered.
  - Click the **Next** button to start the configuration.
12. A status bar is displayed indicating the progress of each component file transfer. The installation may take several minutes.
13. A window is displayed stating that setup is installing the SSL run time. When the SSL run time installation has completed, the window closes automatically.
14. A window displays stating that the Component Broker installation has completed. When you exit this window, your computer automatically restarts. Optionally in this window you can set check boxes to:
- Automatically launch the configuration tool when the computer restarts
  - Automatically display the readme file when the computer restarts

Click the **Finish** button to restart your computer.

## Configuring the Application Server

Before beginning the configuration, verify the following:

- The JDK 1.1.6 classes.zip is included in the CLASSPATH environment variable before any previous versions of the JDK.
- The JDK 1.1.6 bin directory is in the PATH environment variable.

Perform the following steps to configure your Application Server installation:

1. If your computer starts the configuration tool automatically when you restart your computer after performing the installation, skip to the next step. To start the configuration tool from the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Configuration tool**. The Configure New Install window opens.

2. In the Configure New Install window:

- a. Select the **New Install Configuration** radio button to configure your entire Component Broker installation.
- b. Select the **Next** button to continue.

3. **If the DB2 client is installed**

In the Enter Remote DB2 Server Information window:

- a. Type the DB2 user ID used to access the remote DB2 server in the **User ID** field.
- b. Type the password for the DB2 user ID in the **Password** field.
- c. Specify the fully-qualified host name of the server in the **Server Host** field. For example:  
`db2srvr1.austin.ibm.com`
- d. Type the node for this computer in the **Local Node** field.
- e. Click the **Next** button to continue.

4. In the Enter Client's Bootstrap Host Name and Port Number window:

- a. Type the fully-qualified client bootstrap **Host Name** (for example, myhost.austin.ibm.com). The default host name is the local host.

**Note:** The bootstrap host computer contains information about the name servers. Any Component Broker server can act as a bootstrap host.

- b. The default **Port ID** is 900. Accept the default or type a different port ID.
- c. Click the **Next** button to continue.

5. In the Enter Component Broker Secure Socket Layer (SSL) Port ID window:

- a. The default port ID is 3004. This is the port ID that Component Broker will use to perform SSL security. Accept the default or supply a different port ID.
- b. Click the **Next** button to continue.

6. In the Input User ID And Password For Component Broker Service window:

- a. Select a user ID from the list or type a user ID for starting the Component Broker service in the **User ID** field.
- b. Type the password associated with the user ID in the **Password** field.
- c. Click the **Next** button to continue.

**Notes:**

- a. If the selected user ID was created by the Administrator, the password entered must be identical to the existing password. If the password entered does not match the existing user ID, System Management configuration will fail. If the existing user does not have the advanced user right of “logon as a service,” the installation program grants this authority automatically.
  - b. When you uninstall, this user ID and password are not deleted. If you reinstall, you must specify this user ID and password or a new user ID and password combination. If the password entered does not match an existing user ID, System Management configuration will fail.
7. In the Enter DCE Cell Administrator User ID and Password window:
- a. Type the DCE cell administrator ID in the **User ID** field. The default ID is `cell_admin`.
  - b. Type the DCE administrator password in the **Password** field.
  - c. Click the **Next** button to continue.
8. In the Enter System Management Host Name and Port Number window:
- a. Type the fully-qualified host name of the computer on which system manager is installed.
  - b. Type a number representing the port the System Manager User Interface will use to communicate with other System Managers and System Management Agents on all the connected computers. The default port ID is 20002. You can accept the default or supply a different port ID.
  - c. Click the **Next** button to continue.
9. In the Verify Configuration Setting window, verify that the items to be installed are correct.
- Click the **Back** button to review or change any information you previously entered.
  - Click the **Next** button to start the configuration.
10. The Configuration Status window displays information about the items being configured. The configuration process creates the `cbconfig.log` file in the `CBroker\service` directory (where *CBroker* represents the directory in which Component Broker is installed) and places all status information into this log file.

**Notes:**

- a. If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.
  - b. If System Management configuration fails with the following error:  

```
SM - Failed to start CBConnector service. ReturnCode = 1069
```

Refer to “Return Code 1069 From System Manager Configuration” on page 142 for information about fixing the problem. After the problem is fixed, you can reconfigure your Component Broker installation.
11. An informational window is displayed stating that the system management configuration completed successfully. Click the **OK** button to continue.

**12. If the DB2 server is installed on this computer**

As part of the configuration process, several windows are opened that show the configuration of DB2 for the Component Broker server:

- After the system management configuration has completed, a status window is displayed stating that Setup has finished launching the DB2 configuration process. Click the **OK** button to continue.
- After the configuration, two DB2 windows remain open (the other windows close themselves).

If these windows show successful messages at the end, close these windows.

If either window contains an error message, double check the following areas:

- Ensure that the correct transaction monitor is configured by opening a DB2 command line processor window. Enter:

```
get dbm cfg
```

and verifying that the transaction processor monitor name is `somtrx1i`. If not, enter:

```
update database manager configuration using TP_MON_NAME somtrx1i
```

- To verify that the CBSRFCDB has been created, enter:

```
list db directory
```

If the CBSRFCDB has not been created, enter:

```
quit
db2 -f x:\CBroker\etc\inscrtdb.db2
```

where `x:\CBroker` is the base directory where Component Broker is installed.

The DB2 configuration accomplishes:

- The set up of the Database Manager configuration `TP_MON_NAME` to `somtrx1i`, the Component Broker transaction monitor.
- The creation of a database CBSRFCDB for Component Broker Reference Collections database.

13. An informational window is displayed stating that the configuration has completed. Click the **OK** button to close the window.

**Note:** If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.

14. Exit the Component Broker Configuration tool. In the Configuration Status window, click either the **Close** button or the **Cancel** button.
15. If you are using a non-English locale, you must configure the Interface Repository for the current user. The pre-populated Interface Repository (IR) database that is installed on a server is compatible only with the English (United States) setting in the Regional Setting selection of the NT Control Panel. If any other regional setting is selected, you must rebuild the IR using that particular regional setting. To rebuild the IR, perform the following steps:
  - a. If not in the desired locale, change to that locale now.
  - b. Copy the files. From a command prompt, enter:
 

```
copy \CBroker\etc\irempty \CBroker\etc\irwork
```
  - c. Rebuild the IR using the new files. From a command prompt, enter:

```
irpop1
```

On completion, enter:

```
irpop2
```

This procedure could take from ten to fifteen minutes to complete. If multiple servers are needed and the populate time is a consideration, the database files from `irwork` may be copied between servers. This should be done before applications are installed on either the source or target server, because installing an application could modify the IR.

Accessing the IR under any setting other than English (United States) without this correction may generate a `SOMDERROR_IRIncoherent` error.

## Installing the Typical Clients Option

Perform the following steps to install the Typical clients option:

1. Uninstall any previous version of Component Broker.
2. Insert the CBConnector Clients compact disc into your CD-ROM drive.
3. Display the contents of the compact disc.
4. Change to the winnt directory.
5. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To install Component Broker, you must click the Setup icon that looks like a computer. The Component Broker for Windows NT splash screen is displayed followed by the Welcome window.
6. On the Welcome window, click the **Next** button. A message is displayed indicating that Setup is searching for Component Broker packages on your system, then the Choose Destination Location window is displayed.
7. In the Choose Destination Location window:
  - a. The default installation directory is CBroker. To change the drive or directory, click the **Browse** button to open the Choose Directory window in which you can specify a different location.
  - b. In the Choose Directory window:
    - 1) Type a directory name.
    - 2) Click the **OK** button to continue.
    - 3) If the specified directory does not exist, Setup displays a window asking if you want the directory to be created. Click the **Yes** button.
  - c. The Choose Destination Location window is displayed again and lists the directory you specified. Click the **Next** button to continue.
8. In the Select Installation Type window:
  - a. Select the Typical clients installation option.
  - b. Click the **Next** button to continue.

### Notes:

- a. The installation program verifies that your selections are compatible. If they are not compatible, the program displays a message that identifies the conflict.
  - b. If you install the ActiveX client or the Java client for development, remember to install the ActiveX or Java client SDK respectively.
  - c. If you install the ActiveX client with the C++ client, you need to create a two-compiler environment; for details, see Appendix D, "The Two-Compiler Environment (NT only)" on page 189.
9. In the Verify Configuration Setting window, verify that the items to be installed are correct.
    - Click the **Back** button to review or change any information you previously entered.
    - Click the **Next** button to start the configuration.
  10. A status bar is displayed indicating the progress of each component file transfer. The installation may take several minutes.
  11. A window displays stating that the Component Broker installation has completed. When you exit this window, your computer automatically restarts. Optionally in this window you can set check boxes to:
    - Automatically launch the configuration tool when the computer restarts
    - Automatically display the readme file when the computer restarts

Click the **Finish** button to restart your computer.

## Configuring the Typical Clients

Before beginning the configuration, verify the following:

- The JDK 1.1.6 classes.zip is included in the CLASSPATH environment variable before any previous versions of the JDK.
- The JDK 1.1.6 bin directory is in the PATH environment variable.

Perform the following steps to configure your Typical Clients installation:

1. If your computer starts the configuration tool automatically when you restart your computer after performing the installation, skip to the next step. To start the configuration tool from the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Configuration tool**. The Configure New Install window opens.
2. In the Configure New Install window:
  - a. Select the **New Install Configuration** radio button to configure your entire Component Broker installation.
  - b. Select the **Next** button to continue.
3. In the Enter Client's Bootstrap Host Name and Port Number window:
  - a. Type the fully-qualified client bootstrap **Host Name** (for example, myhost.austin.ibm.com).  
**Note:** The bootstrap host computer contains information about the name servers. Any Component Broker server can act as a bootstrap host.
  - b. The default **Port ID** is 900. Accept the default or type a different port ID.
  - c. Click the **Next** button to continue.
4. In the Input User ID And Password For Component Broker Service window:
  - a. Select a user ID from the list or type a user ID for starting the Component Broker service in the **User ID** field.
  - b. Type the password associated with the user ID in the **Password** field.
  - c. Click the **Next** button to continue.

### Notes:

- a. If the selected user ID was created by the Administrator, the password entered must be identical to the existing password. If the password entered does not match the existing user ID, System Management configuration will fail. If the existing user does not have the advanced user right of "logon as a service," the installation program grants this authority automatically.
  - b. When you uninstall, this user ID and password are not deleted. If you reinstall, you must specify this user ID and password or a new user ID and password combination. If the password entered does not match an existing user ID, System Management configuration will fail.
5. In the Enter System Management Host Name and Port Number window:
    - a. Type the fully-qualified host name of the computer on which system manager is installed.
    - b. Type a number representing the port the System Manager User Interface will use to communicate with other System Managers and System Management Agents on all the connected computers. The default port ID is 20002. You can accept the default or supply a different port ID.
    - c. Click the **Next** button to continue.
  6. In the Verify Configuration Setting window, verify that the items to be installed are correct.



- Click the **Back** button to review or change any information you previously entered.
  - Click the **Next** button to start the configuration.
7. The Configuration Status window displays information about the items being configured. The configuration process creates the cbconfig.log file in the *CBroker*\service directory (where *CBroker* represents the directory in which Component Broker is installed) and places all status information into this log file.
 

**Note:** If System Management configuration fails with the following error:

```
SM - Failed to start CBConnector service. ReturnCode = 1069
```

Refer to “Return Code 1069 From System Manager Configuration” on page 142 for information about fixing the problem. After the problem is fixed, you can reconfigure your Component Broker installation.
  8. An informational window is displayed stating that the system management configuration completed successfully. Click the **OK** button to continue.
 

**Note:** If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.
  9. An informational window is displayed stating that the configuration has completed. Click the **OK** button to close the window.
 

**Note:** If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.
  10. Exit the Component Broker Configuration tool. In the Configuration Status window, click either the **Close** button or the **Cancel** button.

## Installing the Custom Install Option

Perform the following steps to install the Custom Install option:

1. If installing a Component Broker server, change to the desired locale before continuing. Once the server is installed, you should not switch to a different locale unless you plan to uninstall the server.
2. Insert the CBConnector compact disc or the CBConnector Clients compact disc into your CD-ROM drive.
3. Display the contents of the compact disc.
4. If installing from the CBConnector Clients compact disc, change to the winnt directory.
5. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To install Component Broker, you must click the Setup icon that looks like a computer. The Component Broker for Windows NT splash screen is displayed followed by the Welcome window.
6. On the Welcome window, click the **Next** button. A message is displayed indicating that Setup is searching for Component Broker packages on your system, then the Choose Destination Location window is displayed.
7. In the Choose Destination Location window:
  - a. The default installation directory is CBroker. To change the drive or directory, click the **Browse** button to open the Choose Directory window in which you can specify a different location.
  - b. In the Choose Directory window:
    - 1) Type a directory name.
    - 2) Click the **OK** button to continue.

- 3) If the specified directory does not exist, Setup displays a window asking if you want the directory to be created. Click the **Yes** button.
  - c. The Choose Destination Location window is displayed again and lists the directory you specified. Click the **Next** button to continue.
8. In the Select Installation Type window:
    - a. Select the Custom Install installation option.
    - b. Click the **Next** button to continue.
  9. In the Select a List of packages for Custom Install window:
    - a. Set the check box next to each piece of software to be installed.
    - b. Click the **Next** button.

**Notes:**

- a. The installation program verifies that your selections are compatible. If they are not compatible, the program displays a message that identifies the conflict.
  - b. If DCE 2.0 is not installed, the install program asks if you want to install it. DCE is a prerequisite for Component Broker.
    - Click the **Yes** button to stop the Component Broker install. After the installation stops, you can install DCE. After installing DCE, restart the Component Broker for Windows NT installation.
    - Click the **No** button to choose another installation selection.
  - c. If you install the ActiveX client or the Java client for development, remember to install the ActiveX or Java client SDK respectively.
  - d. If you install the ActiveX client with the C++ client, you need to create a two-compiler environment; for details, see Appendix D, “The Two-Compiler Environment (NT only)” on page 189.
10. In the Enter SSL Runtime Install Path window:
    - a. The default installation directory for the SSL run time is GSK. To change the drive or directory, specify a different directory in the **Install Path** field.

**Note:** Avoid installing the SSL run time under the Component Broker directory. If the SSL run time is also registered to be used by other products, the cleanup of the Component Broker directory will fail.
    - b. Click the **Next** button to continue.

**Note:** This window does not display if the SSL run time has been installed previously on the system.

**11. If installing the online documentation**

In the Enter NetQuestion Install Path window:

- a. The default installation directory for NetQuestion is IMNNQ. To change the drive or directory, specify a different directory in the **Install Path** field.
- b. Click the **Next** button to continue.

**Note:** This window does not display if NetQuestion has previously been installed on the system. For example, if you have the DB2 V.5 documentation installed, NetQuestion is found in the `x:\imnnq_nt` directory (where `x`: is your DB2 installation drive). The Component Broker installation copies additional files to this location.

**12. If installing the Component Broker server**

In the Select Application Adaptor window:

- a. Verify that the DB2 check box is set.

- b. Click the **Next** button to continue.
13. In the Verify Configuration Setting window, verify that the items to be installed are correct.
    - Click the **Back** button to review or change any information you previously entered.
    - Click the **Next** button to start the configuration.
  14. A status bar is displayed indicating the progress of each component file transfer. The installation may take several minutes.
  15. A window is displayed stating that setup is installing the SSL run time. When the SSL run time installation has completed, the window closes automatically.
  16. **If installing the online documentation**  
 A window is displayed stating that the setup program is installing the NetQuestion search engine. When the NetQuestion installation has completed, the window closes automatically.
  17. A window displays stating that the Component Broker installation has completed. When you exit this window, your computer automatically restarts. Optionally in this window you can set check boxes to:
    - Automatically launch the configuration tool when the computer restarts
 

**Note:** This check box is displayed only if at least one of the following installable units was installed:

      - Server
      - C++ client
      - ActiveX client
      - System Manager
      - System Manager User Interface
      - System Management Agent
    - Automatically display the readme file when the computer restarts

Click the **Finish** button to restart your computer.

## Configuring the Custom Install

Before beginning the configuration, verify the following:

- The JDK 1.1.6 classes.zip is included in the CLASSPATH environment variable before any previous versions of the JDK.
- The JDK 1.1.6 bin directory is in the PATH environment variable.

Depending on the packages you installed, certain of the following steps may not pertain to you. Information at the beginning of each step indicates the conditions under which the step must be performed. Perform the following steps to configure your Custom Install installation:

1. If your computer starts the configuration tool automatically when you restart your computer after performing the installation, skip to the next step. To start the configuration tool from the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Configuration tool**. The Configure New Install window opens.
2. In the Configure New Install window:
  - a. Select the **New Install Configuration** radio button to configure your entire Component Broker installation.
  - b. Select the **Next** button to continue.
3. **If the DB2 client and the Component Broker server are installed**  
 In the Enter Remote DB2 Server Information window:

- a. Type the DB2 user ID used to access the remote DB2 server in the **User ID** field.
- b. Type the password for the DB2 user ID in the **Password** field.
- c. Specify the fully-qualified host name of the server in the **Server Host** field. For example:  
`db2srvr1.austin.ibm.com`
- d. Type the node for this computer in the **Local Node** field.
- e. Click the **Next** button to continue.

#### 4. If you installed the C++ or ActiveX clients

In the Enter Client's Bootstrap Host Name and Port Number window:

- a. Type the fully-qualified client bootstrap **Host Name** (for example, `myhost.austin.ibm.com`). The default host name is the local host.  
**Note:** The bootstrap host computer contains information about the name servers. Any Component Broker server can act as a bootstrap host.
- b. The default **Port ID** is 900. Accept the default or type a different port ID.
- c. Click the **Next** button to continue.

#### 5. If the server is installed

In the Enter Component Broker Secure Socket Layer (SSL) Port ID window:

- a. The default port ID is 3004. This is the port ID that Component Broker will use to perform SSL security. Accept the default or supply a different port ID.
- b. Click the **Next** button to continue.

#### 6. If you installed the System Manager or System Management Agent

In the Input User ID And Password For Component Broker Service window:

- a. Select a user ID from the list or type a user ID for starting the Component Broker service in the **User ID** field.
- b. Type the password associated with the user ID in the **Password** field.
- c. Click the **Next** button to continue.

#### Notes:

- a. If the selected user ID was created by the Administrator, the password entered must be identical to the existing password. If the password entered does not match the existing user ID, System Management configuration will fail. If the existing user does not have the advanced user right of "logon as a service," the installation program grants this authority automatically.
- b. When you uninstall, this user ID and password are not deleted. If you reinstall, you must specify this user ID and password or a new user ID and password combination. If the password entered does not match an existing user ID, System Management configuration will fail.

#### 7. If you installed the Component Broker server

In the Enter DCE Cell Administrator User ID and Password window:

- a. Type the DCE cell administrator ID in the **User ID** field. The default ID is `cell_admin`.
- b. Type the DCE administrator password in the **Password** field.
- c. Click the **Next** button to continue.

#### 8. In the Enter System Management Host Name and Port Number window:

- a. Type the fully-qualified host name of the computer on which system manager is installed. If System Manager is installed on this computer, the default is your local host. Otherwise no default is displayed.
  - b. Type a number representing the port the System Manager User Interface will use to communicate with other System Managers and System Management Agents on all the connected computers. The default port ID is 20002. You can accept the default or supply a different port ID.
  - c. Click the **Next** button to continue.
9. In the Verify Configuration Setting window, verify that the items to be installed are correct.
- Click the **Back** button to review or change any information you previously entered.
  - Click the **Next** button to start the configuration.
10. The Configuration Status window displays information about the items being configured. The configuration process creates the cbconfig.log file in the *CBroker*\service directory (where *CBroker* represents the directory in which Component Broker is installed) and places all status information into this log file.

**Notes:**

- a. If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.
  - b. If System Management configuration fails with the following error:
 

```
SM - Failed to start CBConnector service. ReturnCode = 1069
```

Refer to “Return Code 1069 From System Manager Configuration” on page 142 for information about fixing the problem. After the problem is fixed, you can reconfigure your Component Broker installation.
11. An informational window is displayed stating that the system management configuration completed successfully. Click the **OK** button to continue.
12. **If the Component Broker server is installed and if the DB2 server is installed on this computer**

As part of the configuration process, several windows are opened that show the configuration of DB2 for the Component Broker server:

- After the system management configuration has completed, a status window is displayed stating that Setup has finished launching the DB2 configuration process. Click the **OK** button to continue.
- After the configuration, two DB2 windows remain open (the other windows close themselves).

If these windows show successful messages at the end, close these windows.

If either window contains an error message, double check the following areas:

- Ensure that the correct transaction monitor is configured by opening a DB2 command line processor window. Enter:

```
get dbm cfg
```

and verifying that the transaction processor monitor name is somtrx1i. If not, enter:

```
update database manager configuration using TP_MON_NAME somtrx1i
```

- To verify that the CBSRFCDB has been created, enter:

```
list db directory
```

If the CBSRFCDB has not created, enter:

```
quit
db2 -f x:\CBroker\etc\inscrtdb.db2
```

where x:\CBroker is the base directory where Component Broker is installed.

The DB2 configuration accomplishes:

- The set up of the Database Manager configuration TP\_MON\_NAME to somtrx1i, the Component Broker transaction monitor.
- The creation of a database CBSRFCDB for Component Broker Reference Collections database.

13. An informational window is displayed stating that the configuration has completed. Click the **OK** button to close the window.

**Note:** If the configuration fails, refer to “Reconfiguring Your Component Broker Installation” on page 113 for information about reconfiguring your Component Broker installation.

14. Exit the Component Broker Configuration tool. In the Configuration Status window, click either the **Close** button or the **Cancel** button.

15. If you are using a non-English locale, you must configure the Interface Repository for the current user. The pre-populated Interface Repository (IR) database that is installed on a server is compatible only with the English (United States) setting in the Regional Setting selection of the NT Control Panel. If any other regional setting is selected, you must rebuild the IR using that particular regional setting. To rebuild the IR, perform the following steps:

a. If not in the desired locale, change to that locale now.

b. Copy the files. From a command prompt, enter:

```
copy \CBroker\etc\irempty \CBroker\etc\irwork
```

c. Rebuild the IR using the new files. From a command prompt, enter:

```
irpop1
```

On completion, enter:

```
irpop2
```

This procedure could take from ten to fifteen minutes to complete. If multiple servers are needed and the populate time is a consideration, the database files from irwork may be copied between servers. This should be done before applications are installed on either the source or target server, because installing an application could modify the IR.

Accessing the IR under any setting other than English (United States) without this correction may generate a SOMDERROR\_IRIncoherent error.

---

## Installing the Development Environment

This installation option adds the Component Broker application development environment to an existing Component Broker system. This procedure installs the following:

- CBToolkit
- Server SDK
- C++ Client SDK
- ActiveX Client SDK
- Java Client SDK
- CICS and IMS Application Adaptor SDK
- Samples

You can have difficulties installing JIT 1.1.6 if you installed another version of the JDK. The JIT install program detects whether an incorrect JDK/JRE is found for the upgrade. To resolve this difficulty, use the Registry Editor to change the following:

1. Find the key:

\\HKEY\_LOCAL\_MACHINE\SOFTWARE\JavaSoft\Java Development Kit\1.1

2. Edit the value JavaHome and change to the path to where you installed the JDK 1.1.6 (for example, x:\jdk1.1.6).
3. Edit the value MicroVersion and change to 6.

**Note:** A JDK is required for the ActiveX client. Both the Microsoft C++ compiler and a JDK are required.

To add the components to an existing system:

1. Insert the CBToolkit compact disc into your CD-ROM drive.
2. Display the contents of the compact disc.
3. Start the installation by clicking the Setup icon. The Component Broker for Windows NT splash screen is displayed followed by the Welcome window.
4. Click the **Next** button. A message is displayed indicating that Setup is searching for Component Broker packages on your system.
5. An information window is displayed listing the software already installed.
  - a. Verify the software.
  - b. Click the **OK** button.
6. In the Select a List of Package(s) for Custom Install window:
  - a. Click the **List of packages not yet installed on this system** radio button.
  - b. Click the **Next** button.
  - c. Set the check box next to the software to be installed.
  - d. Click the **Next** button to continue.

**Note:** The installation program verifies that your selections are compatible. If they are not compatible, the installation program displays a message identifying the conflict and allows you to select software that does not conflict. Additionally, the installation program displays a message if there are other requirements or prerequisites required.

7. If VisualAge C++ is not already installed on your system, a warning is displayed indicating that VisualAge C++ is required to do development. VisualAge C++ is a prerequisite for Component Broker development. Click the **OK** button to continue.
8. If one or more of the following components were installed the Enter Port ID for Remote Debugger window is displayed:
  - CBToolkit
  - Server SDK
  - C++ Client SDK
  - ActiveX Client SDK
  - Java Client SDK
  - Samples

In the Enter Port ID for Remote Debugger window, perform the following steps:

- a. Accept the default ID of 8001 or type the appropriate port ID.
  - b. Click the **Next** button to continue.
9. In the Check Setup Information window, verify that the items to be installed are correct.
    - Click the **Back** button to review or change any information you previously entered.
    - Click the **Next** button to continue.
  10. A status bar is displayed indicating the progress of each component file transfer. The installation may take several minutes.

11. In the Setup Complete window, click the **Finish** button to restart your computer.
12. After your computer reboots, to prepare Visual Age for Java 2.0 for Java client application development, certain Component Broker and application files must be imported into Visual Age for Java.
  - a. Open the Visual Age for Java Workframe and create a project named IBM Component Broker.
  - b. On the **File** menu, select **Import**. Specify `X:\CBroker\lib\somajor.zip` and the project defined in the previous step.
  - c. Create a project to contain the Component Broker project Java Client Binding (JCB) jar files.
  - d. On the **File** menu, select **Import**. Specify the JCB files created during the project make step and the project defined in step 12c. The jar files are located in the `Working\Platform\JCB` directory where *Platform* is NT, AIX, and so on.
  - e. On the **File** menu, select **Import**. Specify any project-specific files you have emitted from Object Builder.

**Note:** This step is to import jar/zip/java files that support the specific client development in progress. This step is required for any VAJ project and is not specific to Component Broker.

This step is to import jar/zip/java files that support the specific client development in progress. This step is required for any VAJ project and is not specific to Component Broker.

- f. Open the **Properties** page of the main program and add the classpaths for the IBM Component Broker project and the specific project defined in step 12c.

See the Visual Age for Java 2.0 documentation for more detailed instructions on Visual Age for Java usage.

#### Notes:

1. If you installed the Java client SDK, remember that to run a Java application from the directory in which it was developed, this directory needs to be added to the classpath. In the classpath statement, the current directory is represented by a period (“.”). For information about the classpath and the items that must be listed, see Appendix F, “File Locations and the Java Classpath” on page 199.
2. If the ActiveX client SDK is installed before Microsoft Visual C++, the `MS_INC_DIR` environment variable needs to be defined. Define the environment variable to the path name where the `MSVC++ include` directory is located.

The path name cannot contain the space character and the `MS_INC_DIR` variable should be defined in 8.3 format. If Microsoft Visual C++ was installed using a longer format, modify the `MS_INC_DIR` variable to the short format.

---

## Applying the Component Broker PTF

This section details the steps required to apply the Component Broker PTF. The required steps are:

- “Stop the Component Broker Servers” on page 111
- “Apply the Component Broker PTF” on page 111
- “Restart the System Management Service” on page 111

**Note:** This procedure is only required when a Component Broker PTF is supplied to you.



## Stop the Component Broker Servers

Before applying the Component Broker PTF, stop all running application servers, name servers, and daemons by performing the following steps:

1. Start the System Management User Interface.
2. Become an Expert User. From the **View** menu, select **User Level** → **Expert**.
3. Expand Host Images.
4. For each hostname listed under Host Images:
  - a. Open the pop-up menu for the host.
  - b. Select **Quiesce** to stop the application servers, name servers, and daemons.
5. Exit the System Management User Interface.

## Apply the Component Broker PTF

After all running application servers, name servers, and daemons have been stopped, you can apply the Component Broker PTF. Perform the following steps:

1. Insert the CBConnector Service Pack CD into the CD-ROM drive.
2. Change to \windows directory on the CD drive. Enter:

```
cd \windows
```
3. Run the setup.exe program. Enter:

```
setup
```

The Welcome window is displayed.
4. In the Welcome window, select the **Next** button to continue. A dialog is displayed asking if you would like to apply the fix pack. Select the **Yes** button to continue.
5. If one or more files to be replaced as part of the fix pack are being used during the file transfer, for each file a message box is displayed asking if the file should be overwritten. Select the **Yes** button to continue.
6. A dialog is displayed stating that the fix pack has been installed, and asking if you would like to reboot your computer. Select the **Yes** button to reboot your computer.

## Restart the System Management Service

After the Component Broker PTF is applied, you can restart the application servers, name servers, and daemons. Perform the following steps:

1. Start the System Management User Interface.
2. Become an Expert User. From the **View** menu, select **User Level** → **Expert**.
3. Expand Host Images.
4. For each hostname listed under Host Images:
  - a. Open the pop-up menu for the host.
  - b. Select **Resume** to start the application servers, name servers, and daemons.
5. Exit the System Management User Interface.

---

## Accessing the Online Documentation

To view the online documentation, from the **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Information Library**. Your default browser should open to the documentation homepage, located at the following URL:

`http://localhost:49213/cgi-bin/cbwebx.exe/en_US/cbdoc/Extract/0/index.htm`

If you have difficulty accessing the online documentation, consult the documentation readme (doctread.txt), located in your *CBroke\doc* directory.

To access HTML documentation for the ORB and object service bindings, execute the self-extracting ZIP file `apidocs.exe`, located in the `x:\CBroke\doc\client\javacl` directory. Browse the `packages.html` file to view a listing of available Java packages for the Java client API.

---

## Adding Components to an Existing Installation

Use this procedure to install additional components to an existing Component Broker installation.

1. Depending on the component you want to add, insert either the CBConnector for Windows NT compact disc, the CBConnector Clients compact disc, or the CBToolkit compact disc into your CD-ROM drive.
2. Display the contents of the compact disc.
3. If installing from the CBConnector Clients compact disc, change to the `winnt` directory.
4. Start the installation by double-clicking the Setup icon. The Component Broker for Windows NT splash screen is displayed followed by the Welcome window.
5. Click the **Next** button. A message is displayed indicating that Setup is searching for Component Broker packages on your system.
6. An information window is displayed listing Component Broker components already installed on this system. Click the **OK** button to continue.
7. In the Select a List of Packages for Custom Install window:
  - a. Click the **List of packages not yet installed on the system** radio button.
  - b. Click the **Next** button to refresh the screen.
  - c. Select the check box next to each component you want to install.
  - d. Click the **Next** button to continue.
8. In the Check Setup Information window, verify that the items you specified to be installed are correct. To review or change any information you previously provided, click the **Back** button; otherwise, click the **Next** button. A status bar is displayed showing the file transfer status.
9. A window displays stating that the Component Broker installation has completed. When you exit this window, your computer automatically restarts. Optionally in this window you can set a check box to automatically launch the configuration tool when the computer restarts. Click the **Finish** button to restart your computer.

**Note:** The check box to automatically launch the configuration tool when the computer restarts is displayed only if you have added packages that need to be configured.

---

## Reconfiguring Your Component Broker Installation

Before beginning the reconfiguration, verify the following:

- The JDK 1.1.6 classes.zip is included in the CLASSPATH environment variable before any previous versions of the JDK.
- The JDK 1.1.6 bin directory is in the PATH environment variable.

Depending on the packages you have installed, certain of the following steps may not pertain to you. Information at the beginning of each step indicates the conditions under which the step must be performed.

**Note:** If you experience problems performing the reconfiguration, see “Manually Clearing Up the Existing Configuration” on page 140 before running the Component Broker configuration tool again. You should follow those instructions only as a last resort.

Perform the following steps to reconfigure your Component Broker installation:

1. Start the System Management User Interface.
2. Become an Expert User. From the **View** menu, select **User Level** → **Expert**.
3. Expand Host Images.
4. For each hostname listed under Host Images:
  - a. Open the pop-up menu for the host.
  - b. Select **Quiesce** to stop the application servers, name servers, and daemons.
5. Exit the System Management User Interface.
6. Verify that the CBConnector service is running. From the **Start** menu, select **Settings** → **Control Panel** → **Services**.
7. To start the configuration tool from the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Configuration tool**.
8. If you have never successfully configured Component Broker, the Configure New Install window may be displayed. Otherwise, the Reconfigure window is displayed.

### In the Configure New Install window

- a. Select the **Custom Configuration** radio button.
- b. Select the **Next** button to continue.
- c. In the Reconfiguration window, select the **Custom Reconfiguration** radio button.
- d. Select the **Next** button to continue.

### In the Reconfigure window

- a. Select the **Redo Current Configuration** radio button to reconfigure your entire Component Broker installation or select the **Custom Reconfiguration** radio button to select specific packages to reconfigure.

#### Notes:

- 1) Selecting the **Redo Current Configuration** radio button removes the existing configuration including the existing CDS.
  - 2) If the previous configuration failed, you must select the **Custom Reconfiguration** radio button for your reconfiguration. By default, any packages configured in the current configuration are selected.
- b. Select the **Next** button to continue.

9. If you are reconfiguring specific packages, in the Custom Reconfiguration window:
  - a. Select one or more packages to configure.
  - b. Select the **Overwrite existing configuration** if you want to replace your existing configuration with the new configuration.

**Note:** Selecting this option removes the existing configuration including the existing CDS. Select this option if your previous configuration failed.
  - c. Click the **Next** button to continue.

**10. If the DB2 Client is installed and you are reconfiguring the server**

In the Enter Remote DB2 Server User ID and Password window:

- a. Type the fully-qualified name of the host where the DB2 server is installed.
- b. Type the DB2 node name used locally. The default is db2node. Accept the default or provide a different name.
- c. Type the user ID known to the remote DB2 server.
- d. Type the password to the DB2 server for the specified user.
- e. Type the password again to confirm.

**11. If you are reconfiguring the C++ client or the ActiveX client**

In the Enter Client's Bootstrap Host Name and Port Number window:

- a. Type the fully-qualified client bootstrap **Host Name** (for example, myhost.austin.ibm.com). If the server is installed on the same computer, the default host name is the local host. Otherwise no default is displayed.

**Note:** The bootstrap host computer contains information about the name servers. Any Component Broker server can act as a bootstrap host.
- b. The default **Port ID** is 900. Accept the default or type a different port ID.
- c. Click the **Next** button to continue.

**12. If you are reconfiguring the server**

In the Enter Component Broker Secure Socket Layer (SSL) Port ID window:

- a. The default port ID is 3004. This is the port ID that Component Broker will use to perform SSL security. Accept the default or supply a different port ID.
- b. Click the **Next** button to continue.

**13. If you are configuring the System Manager or the System Management Agent**

In the Input User ID And Password For Component Broker Service window:

- a. Select a user ID from the list or type a user ID for starting the Component Broker service in the **User ID** field.
- b. Type the password associated with the user ID in the **Password** field.
- c. Click the **Next** button to continue.

**Notes:**

- a. If the selected user ID was created by the Administrator, the password entered must be identical to the existing password. If the password entered does not match the existing user ID, System Management configuration will fail. If the existing user does not have the advanced user right of "logon as a service," the installation program grants this authority automatically.
- b. When you uninstall, this user ID and password are not deleted. If you reinstall, you must specify

this user ID and password or a new user ID and password combination. If the password entered does not match an existing user ID, System Management configuration will fail.

**14. If you are reconfiguring the server or the System Manager or if the configuration you are replacing includes System Manager**

In the Enter DCE Cell Administrator User ID and Password window:

- a. Type the DCE cell administrator ID in the **User ID** field. The default ID is `cell_admin`.
- b. Type the DCE administrator password in the **Password** field.
- c. Click the **Next** button to continue.

**15. In the Enter System Management Host Name and Port Number window:**

- a. Type the fully-qualified host name of the computer on which System Manager is installed. If System Manager is installed on your computer, the default is your local host. Otherwise no default is displayed.
- b. Type a number representing the port the System Manager User Interface will use to communicate with other System Managers and System Management Agents on all the connected computers. The default port ID is 20002. You can accept the default or supply a different port ID.
- c. Click the **Next** button to continue.

**16. In the Verify Configuration Setting window, verify that the items to be installed are correct.**

- Click the **Back** button to review or change any information you previously entered.
- Click the **Next** button to start the reconfiguration.

**17. If you selected the Redo Current Configuration or Overwrite existing configuration check boxes, a confirmation dialog is displayed to confirm that you want to remove the existing configuration including CDS. Click the Yes button to start the reconfiguration.**

**18. The Configuration Status window displays information about the items being configured.**

Reconfiguration messages are appended to the `cbconfig.log` file in the `CBroker\service` directory (where `CBroker` represents the directory in which Component Broker is installed).

**Note:** If System Management configuration fails with the following error:

```
SM - Failed to start CBConnector service. ReturnCode = 1069
```

Refer to "Return Code 1069 From System Manager Configuration" on page 142 for information about fixing the problem. After the problem is fixed, you can reconfigure your Component Broker installation.

**19. An informational window is displayed stating that the system management configuration completed successfully. Click the OK button to continue.**

**20. If the DB2 server is installed on this computer and you are reconfiguring the server**

As part of the configuration process, several windows are opened as part of the configuration process that show the configuration of DB2 for the Component Broker server:

- After the system management configuration has completed, a status window is displayed stating that Setup has finished launching the DB2 configuration process. Click the **OK** button to continue.
- After the configuration, two DB2 windows remain open (the other windows close themselves).

If these windows show successful messages at the end, close these windows.

If either window contains an error message, double check the following areas:

- Ensure that the correct transaction monitor is configured by opening a DB2 command line processor window. Enter:

```
get dbm cfg
```

and verifying that the transaction processor monitor name is somtrx1i. If not, enter:

```
update database manager configuration using TP_MON_NAME somtrx1i
```

- To verify that the CBSRFCDB has been created, enter:

```
list db directory
```

If the CBSRFCDB has not been created, enter:

```
quit  
db2 -f x:\CBroker\etc\inscrtdb.db2
```

where x:\CBroker is the base directory where Component Broker is installed.

The DB2 configuration accomplishes:

- The set up of the Database Manager configuration TP\_MON\_NAME to somtrx1i, the Component Broker transaction monitor.
- The creation of a database CBSRFCDB for Component Broker Reference Collections database.

21. An informational window is displayed stating that the configuration has completed. Click the **OK** button to close the window.
22. Exit the Component Broker Configuration tool. In the Configuration Status window, click either the **Close** button or the **Cancel** button.
23. If you reconfigured the Component Broker server, you may need to reconfigure the Interface Repository for the current user. The pre-populated Interface Repository (IR) database that is installed on a server is compatible only with the English (United States) setting in the Regional Setting selection of the NT Control Panel. If any other regional setting is selected, you must rebuild the IR using that particular regional setting. To rebuild the IR, perform the following steps:

- a. If not in the desired locale, change to that locale now.

- b. Copy the files. From a command prompt, enter:

```
copy \CBroker\etc\irempty \CBroker\etc\irwork
```

- c. Rebuild the IR using the new files. From a command prompt, enter:

```
irpop1
```

On completion, enter:

```
irpop2
```

This procedure could take from ten to fifteen minutes to complete. If multiple servers are needed and the populate time is a consideration, the database files from irwork may be copied between servers. This should be done before applications are installed on either the source or target server, because installing an application could modify the IR.

Accessing the IR under any setting other than English (United States) without this correction may generate a SOMDERROR\_IRIncoherent error.

---

## Uninstalling Component Broker

This uninstall procedure is for uninstalling a single Component Broker server system.

### Notes:

1. If you installed any applications in the same directory as the Component Broker run time, you need to uninstall these applications before uninstalling Component Broker. For details on uninstalling an application, see the *Component Broker Application Development Tools Guide*.
2. If you are uninstalling a System Management Agent from a remote System Manager, see the online documentation.

You can use the following procedure to uninstall Component Broker:

1. Log in to DCE as the cell administrator using the password you specified during DCE configuration. You can log in either from a command prompt using `dce_login` or from the DCE log in option of DCE for Windows NT.
2. Remove all Component Broker-related information from the DCE Cell Directory Service.
  - a. From a system command prompt, enter:

```
rmcdsdir all
```
3. Remove the DCE principals and accounts created for you:
  - a. Open the DCE Director.
  - b. Double click on Users.
  - c. Delete CBConnectSM, CBconnector Name Servers, and CBConnector Application servers.
4. Start the uninstall process. There are two ways:
  - From the IBM Component Broker Program entry, select Uninstall.
  - From the Control Panel, select **Add/Remove Programs**. Select **Component Broker** from the list of installed items on your hard disk.
5. The uninstall program confirms your intentions before uninstalling Component Broker.
  - Select the **Yes** button to continue.
  - Select the **No** button to cancel.
6. Restart your computer after the uninstall. If there are files that the uninstall utility is unable to delete during the uninstall process, a warning message is displayed at the end of the uninstall process. The location of these files and subdirectories is part of the message. You need to manually remove these files and subdirectories after a restart.

---

## Uninstall Failure Cleanup

After an uninstall procedure, the following message can be displayed if you did not terminate all servers before uninstalling:

```
Component Broker is removed from your system.  
However, System Management configuration failed.  
Please read the Quick Beginnings booklet for instructions.
```

If the uninstall failed for any reason, you should check the following areas to ensure that they are cleaned up:

1. Delete CBConnector service:

- a. Double-click the Services icon in the Control Panel folder.
- b. If CBCConnector is listed and started, then stop the service.
- c. If you had to stop the CBCConnector service, from a command prompt, enter:

```
instsrv CBCConnector remove
```

**Note:** The instsrv program is part of the Windows NT Resource Kit.

2. Clean up the NT Registry for Component Broker:

- a. From a command prompt, enter:

```
regedt32
```

The Registry Editor window opens.

- b. From the HKEY\_LOCAL\_MACHINE window, open the SOFTWARE folder.
- c. Open the IBM key folder.
- d. Select the Component Broker folder.
- e. From the Registry Editor menu, select **Edit** → **Delete**.

3. Clean up the NT Registry for SSL Runtime.

**Note:** This step is required only if you have the Component Broker server installed.

- a. From a command prompt, enter:

```
regedt32
```

The registry window opens.

- b. From the HKEY\_LOCAL\_MACHINE window, open the the SOFTWARE folder.
- c. Open the IBM key folder.
- d. Open the GSK folder.
- e. Open the CurrentVersion folder.
- f. Select the Uninstall folder.
- g. Get the path and name of the uninstall executable from the UninstallExe entry.
- h. Select the REGAPPS folder.
- i. If "CBroker: REG\_SZ: Registered" is not the only entry in REGAPPS, SSL is also used by one or more other products.
- j. Select the GSK folder. Take note of the SSL install path in the InstallPath entry.
- k. Select the GSK folder.
- l. From the **Registry Editor** menu, select **Edit** → **Delete**.

4. Uninstall the SSL run time. From a command prompt, enter:

```
path_to_ssl_uninstall_exe CBroker
```

Where the *path\_to\_ssl\_uninstall\_exe* is obtained in step 3.

**Notes:**

- a. This step is required only if you have the Component Broker server installed.
- b. If SSL is used by other products, SSL will not be uninstalled. Only the registered entry for CBroker is deleted.

5. End any Component Broker processes:

- a. Open the Task Manager.
- b. Select the **Processes** tab.



- c. If bgmain.exe is there, select it.
  - d. Select the **End Process** button.
  - e. Repeat for bgcmd.exe, somorbd.exe, and somsrsm.exe.
  - f. Close the Task Manager window.
6. Delete all Component Broker files and directories:
- a. From a system command prompt, change directory to where Component Broker is installed.
  - b. If the SSL run time is installed under the Component Broker directory (the directory was determined in step 3 on page 118) and SSL is used by other products (determined in step 3 on page 118), do not remove the files and directories belonging to SSL located in the Component Broker directory. Delete only Component Broker files and directories.  
  
If SSL is not installed under the Component Broker directory, delete all Component Broker files and directories.
7. Remove all references to Component Broker directories from the environment variables:
- a. Double-click the System icon in the Control Panel folder
  - b. Remove all references to Component Broker directories from the environment variables.
  - c. If SSL run time is uninstalled, also delete all references to SSL directories from the CLASSPATH, NLSPATH, and PATH environment variables.

The uninstall process is complete, you can restart your computer.



---

## Chapter 9. Installing Component Broker for AIX

This chapter contains the following procedures for Component Broker for AIX.

- “Installing the Server Run Time”
- “Installing the Client Run Time” on page 123
- “Configuring the Run Time” on page 126
- “Reconfiguring the Run Time” on page 127
- “Installing the Development Environment” on page 127
- “Applying the Component Broker PTF” on page 128
- “Accessing the Online Documentation” on page 130
- “Uninstalling Component Broker” on page 130

**Note:** Before installing this version of Component Broker for AIX completely remove any previous versions. To determine the installed software on a computer, enter:

```
lslpp -l | grep CB
```

If any Component Broker products are listed, uninstall Component Broker for AIX. For information about uninstalling, see “Uninstalling Component Broker” on page 130.

These procedures assume the korn shell (ksh) as the default shell. Component Broker for AIX supports only the korn shell.

To complete the instructions in this chapter, you must be logged in as root (unless otherwise specified). Additionally, they assume installation, configuration, and uninstallation using the ASCII version of SMIT (using the `smi tty` command from the shell prompt). To use the GUI version of SMIT, use the `smi t` command from the shell prompt and modify the instructions as appropriate.

See the “Installation and Configuration” section of the *Late Breaking News* provided with Component Broker for additional important setup information.

---

### Installing the Server Run Time

This set of installation instructions covers the installation (and included components) for:

- System Manager
  - Component Broker server
  - C++ client
  - System Manager
  - System Manager User Interface
  - DB2 Application Adaptor
  - Documentation
- Standalone Workstation
  - Component Broker server
  - C++ client
  - System Manager
  - System Manager User Interface
  - DB2 Application Adaptor
  - Documentation
- Application Server
  - Component Broker Server
  - C++ client
  - System Management Agent

- DB2 Application Adaptor

**Note:** Avoid performing a Custom Install on an individual Component Broker fileset because there is no configuration entry added to ODM to do the configuration. For a server run time installation you should always install from the CBCConnector.TYPICAL LPP level.

To begin the installation process:

1. If not already logged in as root, do so now.
2. Insert the CBCConnector compact disc into the CD-ROM drive.

**Note:** If a CD-ROM file system does not already exist, you must create one. For information about creating a CD-ROM file system, see “Creating a CD-ROM File System” on page 76.

3. Mount the file system. To mount, enter:

```
mount /cdrom
```

4. At a shell prompt, enter:

```
smitty install_latest
```

The Install and Update from LATEST Available Software screen is displayed.

5. On this screen:
  - a. In the **INPUT device / directory for software** field, press the **F4** key to display a list of available devices from which you can select the appropriate input device (/dev/cd0) and press the **Enter** key. This screen is redisplayed with additional fields.
  - b. In the **SOFTWARE to install** field, press the **F4** key to display a list of the software packages contained on the compact disc.
  - c. In the SOFTWARE to install list, scroll down and select one of the following entries:

**For a Application Server installation**

2.0.0.0 for CBCConnector.TYPICAL (Application\_Server)

**For a Standalone Workstation installation**

2.0.0.0 for CBCConnector.TYPICAL (Standalone\_Workstation)

**For a System Manager installation**

2.0.0.0 for CBCConnector.TYPICAL (System\_Manager\_Server)

- d. Press the **F7** key to select the packages you want to install.
  - e. Press the **Enter** key. The Install and Update from LATEST Available Software screen is redisplayed, but the software packages selected are in the **SOFTWARE to install** field.
  - f. Ensure that the **AUTOMATICALLY install requisite software?** and **EXTEND file systems if space needed?** fields are yes.
  - g. Press the **Enter** key. The ARE YOU SURE? screen is displayed.
6. On the ARE YOU SURE? screen, press the **Enter** key to continue.

The installation process begins. After the files are copied, the command status, as indicated in the upper left corner of the screen should state OK. When you receive this OK status, all files were copied and the system is ready to be configured. Press the **F10** key to exit smitty.

The files installed during the run-time installation are in the /usr/lpp/CBCConnector directory.

## Notes:

1. If any errors occurred, view the \$HOME/smit.log file. If the cause of the error is not clear, save a copy of the smit.log file and report the problem to IBM.
2. Component Broker, while placing most of its files on the /usr filesystem, also uses space from the /var filesystem. You must ensure that you have adequate space available on the /var filesystem when doing an Application Install of Component Broker built applications. Space can be determined for your Component Broker Applications by entering the following command in the location in which you built your application:

```
du -sk *.ddl *.so
```

The total of these files will be the amount of space required for the application. An additional 50MB of space should be allocated beyond that for utilization by Component Broker Systems Management purposes.

You should consider creating a separate filesystem for DCE other than /var. If Component Broker and DCE share the same filesystem and if space in /var is exhausted, you may not be able to perform a dce\_login. In addition, System Manager could terminate.

If Component Broker Load Application fails due to filling up the /var filesystem because of files being copied during load application, then Component Broker Systems Management frees disk space by removing the files that were installed by this failed load application attempt. There are two ways you can recover from this:

- If you have a JFS filesystem and adequate space in the LVM, then you can increase the filesystem space on /var dynamically.
- If you cannot expand your JFS filesystem on /var, but you have space allocated on other filesystems then you could use symbolic links to your aixApps directory to another filesystem. Please refer to the AIX documentation for information regarding symbolic links.

There are additional steps that are required if your Component Broker server is installed on a DB2 client computer. For more information, see "Installing a Component Broker Server on a DB2 Client" on page 173.

---

## Installing the Client Run Time

**Note:** Avoid performing a Custom Install on an individual Component Broker fileset because there is no configuration entry added to ODM to do the configuration. For a client installation you should always install from the CBclient.custom LPP level.

This set of installation instructions covers the installation (and included components) for:

- C++ Client with the System Management Agent
- Java Client with the System Management Agent
- Typical Clients (includes both the C++ and Java clients with the System Management Agent)

To begin the installation process:

1. If not already logged in as root, do so now.
2. Insert the CBConnector Clients compact disc into the CD-ROM drive.

**Note:** If a CD-ROM file system does not already exist, you must create one. For information about creating a CD-ROM file system, see "Creating a CD-ROM File System" on page 76.

3. Mount the file system. To mount, enter:

```
mount /cdrom
```

4. Change to the AIX directory. Enter:

```
cd /cdrom/aix
```

5. At a shell prompt, enter:

```
smitty install_latest
```

The Install and Update from LATEST Available Software screen is displayed.

6. On this screen:

- a. In the **INPUT device / directory for software** field, type a period (.) to represent the current directory and press the **Enter** key. This screen is redisplayed with additional fields.
- b. In the **SOFTWARE to install** field, press the **F4** key to display a list of the software packages contained on the compact disc.
- c. In the SOFTWARE to install list, scroll down and select one of the following:  
**For a C++ client installation** 2.0.0.0 for CBclient.custom (C++\_Client)  
**For a Java client installation** 2.0.0.0 for CBclient.custom (Java\_Client)  
**For a Typical clients installation** 2.0.0.0 for CBclient.custom (Typical\_Clients)
- d. Press the **F7** key to select the packages you want to install.
- e. Press the **Enter** key. The Install and Update from LATEST Available Software screen is redisplayed, but the software packages selected are in the **SOFTWARE to install** field.
- f. Ensure that the **AUTOMATICALLY install requisite software?** and **EXTEND file systems if space needed?** fields are yes.
- g. Press the **Enter** key. The ARE YOU SURE? screen is displayed.

7. On the ARE YOU SURE? screen, press the **Enter** key to continue.

The installation process begins. After the files are copied, the command status, as indicated in the upper left corner of the screen should state OK. When you receive this OK status, all files were copied and the system is ready to be configured. Press the **F10** key to exit smitty.

The files installed during the run-time installation are in the /usr/lpp/CBConnector directory.

#### Notes:

1. If any errors occurred, view the \$HOME/smit.log file. If the cause of the error is not clear, save a copy of the smit.log file and report the problem to IBM.
2. Component Broker, while placing most of its files on the /usr filesystem, also uses space from the /var filesystem. You must ensure that you have adequate space available on the /var filesystem when doing an Application Install of Component Broker built applications. Space can be determined for your Component Broker Applications by entering the following command in the location in which you built your application:

```
du -sk *.ddl *.so
```

The total of these files will be the amount of space required for the application. An additional 50MB of space should be allocated beyond that for utilization by Component Broker Systems Management purposes.

You should consider creating a separate filesystem for DCE other than /var. If Component Broker and DCE share the same filesystem and if space in /var is exhausted, you may not be able to perform a dce\_login. In addition, System Manager could terminate.

If Component Broker Load Application fails due to filling up the /var filesystem because of files being copied during load application, then Component Broker Systems Management frees disk space by removing the files that were installed by this failed load application attempt. There are two ways you can recover from this:

- If you have a JFS filesystem and adequate space in the LVM, then you can increase the filesystem space on /var dynamically.
- If you cannot expand your JFS filesystem on /var, but you have space allocated on other filesystems then you could use symbolic links to your aixApps directory to another filesystem. Please refer to the AIX documentation for information regarding symbolic links.

---

## Configuring Your Component Broker User ID

Before configuring and using Component Broker, you must ensure that the environment is properly configured. Do this by updating the .profile and .kshrc of the Component Broker *cbuser* user ID

### Notes:

1. The following steps are valid for an AIX Korn Shell.
2. Some of these lines might have been added to the .profile during previous steps. They are included here for completeness.

Perform the following steps:

1. Logon as *cbuser*.
2. Ensure the following lines are at the top of the .profile:

```
#!/bin/ksh
START=$HOME/.kshrc
ENV='${START[(($_=-1)+($_=0)-($_!=-$_{-%%*i*})]}'
export START ENV
```

3. Add the following lines to the end of the .profile:

```
PATH=/usr/jdk_base/bin:$PATH
CLASSPATH=/usr/jdk_base/lib/classes.zip:.$CLASSPATH
. /usr/lpp/CBConnector/bin/CBConnector.profile
. $HOME/sql1lib/db2profile
```

4. Edit .dtprofile to uncomment the last line. It should look similar to:

```
DTSOURCEPROFILE=true
```

5. Edit the .kshrc file to add the following as the last line:

```
. /usr/lpp/CBConnector/bin/CBConnector.profile
```

**Note:** If you don't have a .kshrc file in the \$HOME directory, you can create the file by entering the above line in the file.

---

## Configuring the Run Time

### Notes:

1. To configure Component Broker, the system must have a minimum of 400 MB of paging space.
2. The Component Broker install code does not error check return codes from DB2 commands. Anytime you configure or reconfigure Component Broker, you must review the return codes in the smit.log for any DB2 command errors that may have occurred.

To configure this installation from the System Management (SMIT) main screen:

1. If not already logged in as root, do so now.
2. At a shell prompt, enter:

```
smitty apps
```

3. Select the appropriate configuration:

#### To configure a System Manager installation

Configure CBConnector (System\_Manager\_Server)

#### To configure a Application Server installation

Configure CBConnector (Application\_Server)

#### To configure a Standalone Workstation installation

Configure CBConnector (Standalone\_Workstation)

#### To configure a C++ client installation

Configure CBConnector (C++\_Client)

**Note:** Your original installation choice will determine which selections are available to you on this screen. Not all of the selections described here will be displayed on your screen.

4. On the Component Broker Configuration screen.
  - a. In the **Client Bootstrap Host Name** field, type the fully-qualified client bootstrap host name (for example, myhost.austin.ibm.com). On a Standalone\_Workstation or System\_Manager server, this is usually your fully-qualified server name. The bootstrap host contains information about the name servers so that all other computers can get their naming services populated. There is no default for the bootstrap host name. Any Component Broker server can be the bootstrap host. Normally, this is the first Component Broker server; therefore, this computer is your bootstrap host.
  - b. In the **Client Bootstrap Port ID** field, accept the default (900) or type a different port ID.
  - c. If the configuration provides fields for the DCE administrator ID and password, type these values in the appropriate fields.

**Note:** The configuration screens for the C++ client do not contain fields for the DCE Cell Administrator Name or DCE password.
  - d. In the **System Management Port ID** field, accept the default (20002) or type a different port ID. This is the port ID the System Manager User Interface will use to communicate with other System Managers and System Management Agents on all the connected computers.
  - e. In the **System Management Host Name** field, type the fully-qualified host name (for example, myhost.austin.ibm.com). The Component Broker System Manager controls servers, clients, and applications on one or more hosts. Type the host name on which the System Manager is installed. If you enter the local host name, the System Manager is configured locally and the client becomes locally managed. If you specify a remote System Manager host, the System Management Agent is configured locally and is managed by the remote System Manager.



**Note:** Ensure that the remote System Manager host is installed and the System Management service (bgmain) on that system is running before installing the remote System Management Agent.

- f. In the **CBConnector userid** field, type the user name. This name needs to be the user name specified during DB2 configuration.
- g. Ensure that the **Create CDS now?** field is Y. If not, use the Tab key to toggle the value.
- h. Press the **Enter** key to begin configuration. This can take several minutes.

The configuration is completed when the command status, as indicated in the upper left corner of the screen should state OK.

5. Press the **F10** key to exit SMIT.

---

## Reconfiguring the Run Time

The Component Broker run time can be configured any number of times; but, during reconfiguration, any existing CDS will be lost and replaced with the newly created CDS. Whenever Component Broker is reconfigured, existing Component Broker processes will be terminated and applicable Component Broker processes will be started (or restarted). These processes are:

- The ORB daemon (somorbd)
- Any application or name servers (somsrsm)
- The System Management service (bgmain)
- The System Manager User Interface (bhgeui)

Orphaned semaphores may remain from one of these processes, and it is possible that they may prevent proper operation of a Component Broker program. Remove these semaphores using the cleanipc tool. This tool is located at:

```
/usr/lpp/CBConnector/bin/cleanipc.ksh
```

**Note:** The Component Broker install code does not error check return codes from DB2 commands. Anytime you configure or reconfigure Component Broker, you must review the return codes in the smit.log for any DB2 command errors that may have occurred.

---

## Installing the Development Environment

Before installing the development environment, install the Component Broker run-time environment.

To install the Component Broker development environment and the application development tools:

1. If not already logged in as root, do so now.
2. Insert the CBToolkit compact disc into the CD-ROM drive.
3. Mount the file system. To mount, enter:

```
mount /cdrom
```

4. At a shell prompt, enter:

```
smitty install_latest
```

The Install and Update from LATEST Available Software screen is displayed.

5. On this screen:

- a. In the **INPUT device / directory for software** field, press the **F4** key to display a list of available devices of which you can select the appropriate input device (/dev/cd0) and press the **Enter** key. This screen is redisplayed with additional fields.
  - b. In the **SOFTWARE to install** field, press the **F4** key to display a list of the software packages contained on the compact disc.
  - c. In the SOFTWARE to install list, scroll down and select one or more of the following:
    - For a C++ client SDK installation 2.0.0.0 for CBToolkit.TOOLKIT (C++ClientSDK)**
    - For a client samples SDK installation 2.0.0.0 for CBToolkit.TOOLKIT (clientSamplesSDK)**
    - For a Java client SDK installation 2.0.0.0 for CBToolkit.TOOLKIT (JavaClientSDK)**
    - For a server SDK installation 2.0.0.0 for CBToolkit.TOOLKIT (serverSDK)**
  - d. Press **F7** to select this package.
  - e. Press the **Enter** key. The Install and Update from LATEST Available Software screen is redisplayed, but the software packages selected are in the **SOFTWARE to install** field.
  - f. Ensure that the **AUTOMATICALLY install requisite software?** and **EXTEND file systems if space needed?** fields are yes.
  - g. Press the **Enter** key. The ARE YOU SURE? screen is displayed.
6. On the ARE YOU SURE? screen, press the **Enter** key to continue.

The installation process begins. After the files are copied, the command status, as indicated in the upper-left corner of the screen should state OK. When you receive this OK status, all file sets are copied and the system is ready to be configured. Press the **F10** key to exit smitty.

**Note:** If any errors occurred, view the \$HOME/smit.log file in the / file system. If the cause of the error is not clear, save a copy of the smit.log file and report the problem to IBM.

The file sets installed during the development environment installation are in the /usr/lpp/CBToolkit directory.

You can now unmount the CD-ROM drive to eject the compact disc. To unmount the drive, enter:

```
umount /cdrom
```

Where *cdrom* represents the file system.

---

## Applying the Component Broker PTF

This section details the steps required to apply the Component Broker PTF. This procedure is required only when a Component Broker PTF is supplied to you. The required steps are:

- “Stop the System Management Service”
- “Apply the Component Broker PTF” on page 129
- “Restart the System Management Service” on page 130

## Stop the System Management Service

Before applying the Component Broker PTF, stop all running application servers, name servers, daemons, and the System Management service by performing the following steps:

1. Log on as a Component Broker user.
2. Stop the servers and daemons:

- a. Start the System Management User Interface.
  - b. Become an Expert User. From the **View** menu, select **User Level** → **Expert**.
  - c. Expand Host Images.
  - d. For each hostname listed under Host Images:
    - 1) Open the pop-up menu for the host.
    - 2) Select **Quiesce** to stop the application servers, name servers, and daemons.
  - e. Exit the System Management User Interface.
3. Stop the System Management Service (bgmain):
    - a. From a shell prompt, enter:
 

```
smitty apps
```
    - b. Move the cursor to **Stop the System Management Application** and press the **Enter** key.
    - c. On the Stop the System Management Application now? window, press the **Tab** key to change the value in the **Stop the System Management Application now?** field to y and press the **Enter** key.
    - d. Press the **F10** key to exit smitty.

## Apply the Component Broker PTF

After all running application servers, name servers, daemons, and the System Management service have been stopped, you are ready to apply the Component Broker PTF. Perform the following steps:

1. If not already logged in as root, do so now.
2. Insert the CBConnector Service Pack CD into the CD-ROM drive.
3. Mount the file system by entering:
 

```
mount /cdrom
```

**Note:** For information about creating a CD-ROM file system, see “Creating a CD-ROM File System” on page 76.
4. From a shell prompt, enter:
 

```
smitty update_all
```

The Update Install Software to Latest Level (Update All) window is displayed.
5. In this window:
  - a. Type /cdrom/aix in the **INPUT device / directory for software** field.
  - b. Press the **Enter** key to continue.
6. In the next window, press the **Enter** key to display the ARE YOU SURE? window.
7. Press the **Enter** key to commit the install.
8. When the installation is complete, press the **F10** key to exit SMIT.
9. Unmount the CD-ROM. Enter:
 

```
umount /cdrom
```

## Restart the System Management Service

After the Component Broker PTF is applied, you can restart the System Management service and the servers. Perform the following steps:

1. Log on as Component Broker User.
2. Start the System Management service:
  - a. From a shell prompt, enter:

```
smitty apps
```
  - b. Select **Start the System Management Application** and press the **Enter** key.
  - c. On the Start the System Management Application now? window, press the **Tab** key to change the value in the **Start the System Management Application now?** field to **y** and press the **Enter** key.
3. Start the application servers, name servers, and daemons. Perform the following steps:
  - a. Start the System Management User Interface.
  - b. Become an Expert User. From the **View** menu, select **User Level** → **Expert**.
  - c. Expand Host Images.
  - d. For each hostname listed under Host Images:
    - 1) Open the pop-up menu for the host.
    - 2) Select **Resume** to start the application servers, name servers, and daemons.
  - e. Exit the System Management User Interface.

---

## Accessing the Online Documentation

To access the Component Broker online documentation, open your browser and enter the following URL (you may want to bookmark this page for easy reference):

```
http://localhost:49213/cgi-bin/cbwebx.exe/en_US/cbdoc/Extract/0/index.htm
```

If you have difficulty accessing the online documentation, consult the documentation readme (docread.txt), located in your `lpp/CBConnector/doc/en_US` directory.

---

## Uninstalling Component Broker

Before uninstalling Component Broker:

1. Stop the servers and daemons:
  - a. Logon as cbuser.
  - b. Start the System Management User Interface.
  - c. Become an Expert User. From the **View** menu, select **User Level** → **Expert**.
  - d. Expand Host Images.
  - e. For each hostname listed under Host Images:
    - 1) From the pop-up menu for the host, select **Quiesce** to stop the application server, name servers, and daemons.
    - 2) Exit the System Management User Interface.

2. Stop the System Management Service (bgmain):
  - a. From a shell prompt, enter:

```
smitty apps
```
  - b. Move the cursor to **Stop the System Management Application** and press **Enter**.
  - c. On the Stop the System Management Application now? window, press the **Tab** key to change the response to **y** and press **Enter**.

After all the appropriate servers are removed, you can uninstall Component Broker. To uninstall Component Broker:

1. If not already logged in as root, do so now.
2. At a shell prompt, enter:

```
smitty remove
```

The Remove Installed Software screen is displayed.
3. On the Remove Installed Software screen:
  - a. In the **SOFTWARE name** field, type `CBclient` to remove the run-time and development environments.
  - b. In the **PREVIEW only?** field, change to **n**.
  - c. In the **REMOVE dependent software?** field, change to **y**.
  - d. Press the **Enter** key to remove Component Broker for AIX from your system.
4. Press the **F10** key to exit smitty.
5. If you created a `/var/CBConnector` file system before installing Component Broker, you can remove it at this time. The creation of the file system is described in “AIX Configuration Considerations” on page 58. To remove the file system, perform the following steps:
  - a. If not already logged in as root, do so now.
  - b. Unmount the file system. Enter:

```
umount /var/CBConnector
```
  - c. Remove the file system. Enter:

```
smitty rmjfs
```
  - d. For the FILE SYSTEM name, press the **F4** key to select the file system.
  - e. On the FILE SYSTEM name screen, move the cursor to `/var/CBConnector` and press the **Enter** key.
  - f. Set the **Remove Mount Point** to **yes**.
  - g. Press the **Enter** key to remove the file system. The ARE YOU SURE? screen is displayed.
  - h. On the ARE YOU SURE? screen, press the **Enter** key to continue.
  - i. Press the **F10** key to exit smitty.

To remove any semaphores remaining after uninstalling Component Broker, shutdown and reboot your workstation. Enter:

```
shutdown -Fr
```

This command shuts down your workstation and reboots.



---

## Chapter 10. Installing the Component Broker Client for Windows 95

This chapter contains procedures for installing and uninstalling Component Broker for Windows 95. The only installable unit for Component Broker for Windows 95 is the Java client.

---

### Installing the Java Client

Use this procedure to install the Component Broker Java client on a Windows 95 system.

1. Insert the CBConnector compact disc into your CD-ROM drive.
2. Display the contents of the compact disc.
3. Display the contents of the Win95 folder.
4. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To install Component Broker, click the Setup icon that looks like a computer. The Component Broker splash screen is displayed followed by the Welcome window.
5. Click the **Next** button. A message is displayed indicating that Setup is searching for Component Broker packages on your system. Then, the Choose Destination Location window is displayed.
6. In this window:
  - a. The default install directory is x:\CBroker. To change the drive or directory, click the **Browse** button to open the Choose Directory window in which you can specify a different location.
  - b. In the Choose Directory window:
    - 1) Type a directory name.
    - 2) Click the **OK** button to continue.
    - 3) If the directory does not exist, Setup displays a window asking if you want the directory to be created. Click the **Yes** button.
  - c. The Choose Destination Location window is displayed again and lists the directory you specified. Click the **Next** button to continue. The Select Installation Type window is displayed.
7. In this window, the only choice is the Java client. Click the **Next** button to continue. The Check Setup Information window is displayed.
8. In this window, verify that the items to be installed are correct. Because the Java client is the only installable unit, Click the **Next** button to continue. A status bar is displayed indicating the progress of the file transfer.
9. A window displays stating that the Component Broker installation has completed. When you exit this window, your computer automatically restarts. Optionally in this window you can select a check box to automatically display the readme file when the computer restarts. Click the **Finish** button to restart your computer.

---

## Uninstalling the Java Client

Use the following procedure to uninstall the Component Broker Java client from a Windows 95 system.

1. If you installed any applications in the same directory as the Java client, uninstall these applications before uninstalling the Java client.
2. Start the uninstall procedure. You can either:
  - Select **Uninstall** from the IBM Component Broker program entry.
  - Select Component Broker from the list of installed items in the Add/Remove Programs Properties dialog. To access this dialog, select Control Panel → Add/Remove Programs.
3. The uninstall program confirms your intentions before the Java client is uninstalled. Click the **Yes** button to continue; **No** to cancel.
4. Reboot your system. If there are files that the uninstall utility cannot delete, a warning message is displayed. The location and the files that were not deleted are part of this message. You need to manually remove these files and subdirectory.

---

## Recovering from an Uninstall Failure

If the uninstall procedure failed for any reason, you should check the following areas to ensure that they are correctly cleaned up:

1. Clean up the Windows 95 Registry:
  - a. From Windows 95 **Start** button, select **Run**. The Run dialog is displayed.
  - b. In this dialog, enter  
`regedit`  
The Registry Editor window is displayed.
  - c. Expand HKEY\_LOCAL\_MACHINE → SOFTWARE → IBM, and select Component Broker.
  - d. From the Registry Editor menu, select **File** → **Delete**.
2. Delete all Java client files and subdirectories:
  - a. From a system command prompt, change directory to where the Java client was installed.
  - b. Delete all Java client files and directories.
3. Edit the AUTOEXEC.BAT file to remove the line that calls the compBrok.bat file.

The uninstall procedure is complete. Reboot your computer.



---

## Chapter 11. Installing the Component Broker Client for OS/2

This chapter contains procedures for installing and uninstalling Component Broker for OS/2. The only installable unit for Component Broker for OS/2 is the Java client.

---

### Installing the Java Client

Use this procedure to install the Component Broker Java client on an OS/2 operating system.

1. Insert the CBCConnector compact disc into your CD-ROM drive.
2. At an OS/2 command prompt, make the CD-ROM drive your current drive.
3. Change to the OS2 directory. Enter:  

```
cd os2
```
4. Start the installation by entering:  

```
java setup
```

The Component Broker Welcome window is displayed.
5. In the Component Broker Welcome window, click the **Next** button to continue.
6. In the Choose Destination Location window:
  - a. The default install directory is x:\CBroker. To change the drive or directory, click the **Browse** button to open the Choose Directory window in which you can specify a different location.
  - b. In the Choose Directory window:
    - 1) Type a directory name with eight or fewer characters.
    - 2) Click the **OK** button to continue.
    - 3) If the directory does not exist, Setup displays a window asking if you want the directory to be created. Click the **Yes** button.
  - c. The Choose Destination Location window is displayed again and lists the directory you specified. Click the **Next** button to continue.
  - d. In the dialog, select the configuration file to be updated. Click the **Install** button to begin the installation.
7. A status bar is displayed indicating the progress of the file transfer. When the installation completes, the Installation Complete window is displayed. Click the **Finish** button to close the installation window.
8. Reboot your system.

---

### Uninstalling the Java Client

Use the following procedure to uninstall the Component Broker Java client from an OS/2 operating system.

1. If you installed any applications in the same directory as the Java client, uninstall these applications before uninstalling the Java client.
2. At an OS/2 command prompt, change to the drive and directory in which the Java client is installed.
3. Start the uninstall procedure. Enter:  

```
java uninstall
```

4. The uninstall program confirms your intentions before the Java client is uninstalled. Click the **Yes** button to continue; **No** to cancel.
5. A status bar is displayed indicating the progress of the uninstall. When the procedure completes, the Uninstallation Complete window is displayed. Click the **Finish** button to close the uninstallation window.
6. Use the OS/2 Drives Graphical User Interface to remove the installation directory and its contents.
7. Reboot your system.

---

## Chapter 12. Installing the Component Broker Java Client on a Web Server

Use this procedure to install the Component Broker Java client on a Web server as a downloaded Java applet.

You should use this procedure only if you are familiar with the Web server you are using and are familiar with how to install applets on the Web server. You need an unzip program to decompress the file containing the Component Broker Java client.

1. Insert the CBCConnector Clients compact disc into your CD-ROM drive.
2. Change to the platform-specific directory on the CBCConnector Clients compact disc (for example, the `winnt\jclient` directory).
3. Copy the `somojor.zip` file to a target directory on the web server.  
**Note:** The target directory on the web server depends on how you have organized your web page and applet class file structure. The target directory should reside in a directory at the top of the Java package hierarchy used by your applets.
4. Unzip the `somojor.zip` file. The `\COM` and `\ORG` directories are created as subdirectories of the directory into which you copied the zip file.

If the Component Broker server has been configured as an SSL secured server, you must perform the following additional steps to install the downloadable SSL secured Component Broker Java client:

1. Insert the CBCConnector Clients compact disc into your CD-ROM drive.
2. Change to the `\jclient` directory on the CBCConnector Clients compact disc.
3. Copy the `somojse.zip` file to a target directory on the web server. This should be the same directory into which you copied `somojor.zip`.  
**Note:** The target directory on the web server depends on how you have organized your web page and applet class file structure. The target directory should reside in a directory at the top of the Java package hierarchy used by your applets.
4. Unzip the `somojse.zip` file. If prompted to overwrite the files, answer `yes`.  
**Note:** You must define a server keyring for each of the servers that the client will use and you must define a client keyring that contains the trust-basis of the servers for use by each client. The client keyring file is referenced in the client style properties file. Both the client keyring file and the client style properties file must be installed on the web server. For more information on setting up Java applets with SSL, see the "Use of DCE and SSL Based Security in Your Enterprise" topic in the *Component Broker for Windows NT and AIX System Administration Guide*.

To run your Java applications from the directory in which you are working on them, you need to add the current directory ("`.`") to your classpath.

---

## Using a Downloaded Java Client

Downloading a Java client causes the Java ORB classes to be downloaded. Due to Java security, these classes can only open socket connections to the same physical IP address from which the classes were downloaded. This means that ORB requests can only flow to the web server.

Placing all the resources that the Java client needs to access at the same physical IP address as the web server allows the downloaded Java client to fully participate in the Component Broker environment. This means placing the web server, the Component Broker name server, the Component Broker application servers, and all Java classes that comprise the Java applet at the same physical IP address. Delegated requests to third tier resources are supported, as long as the Java client does not have to directly access the third tier resource.

---

## Chapter 13. Problem Determination

This section documents information related to installing or uninstalling Component Broker. Topics in this section include:

- “Understanding Install or Uninstall Failures”
- “Setup Unable to Find Domain Name”
- “Coexisting With the Hummingbird SOCK5 Server” on page 140
- “Manually Clearing Up the Existing Configuration” on page 140
- “Unable to Start the Component Broker Configuration Tool” on page 141
- “Return Code 1069 From System Manager Configuration” on page 142
- “Incomplete PATH After Rebooting Component Broker or Installing Another Product” on page 142
- “Database CBSRFCDB Not Created Correctly” on page 143

---

### Understanding Install or Uninstall Failures

WIN

The Component Broker install and uninstall processes use the Windows NT 16-bit Windows Subsystem. Occasionally, the Component Broker install or uninstall fails when the Windows NT Virtual DOS Machine encounters an illegal instruction. One of the following symptoms appear when this failure occurs:

- A 16 Bit Windows Subsystem message window is generated with the message: Hidden Console of WOW VDM. The NTVDM CPU has encountered an illegal instruction. Choose 'Close' to terminate the application.
- The install or uninstall process starts and ends without the Component Broker Welcome window opening.

For information on resolving this problem, consult the following Microsoft Knowledge Base articles:

- *Q165214 Troubleshooting MS-DOS-Based Programs in Windows NT4.0*
- *Q119365 Runtime Error 6009 – Default Environment Size*

---

### Setup Unable to Find Domain Name

WIN

When running the Setup.exe, you may see the following severe error message:

```
Setup is unable to find the domain name for this system.  
Please add the domain name in TCP/IP protocol. Setup is  
terminated.
```

If you receive the above error message, add the domain name to the TCP/IP protocol in the Network control panel. Perform the following steps:

1. From the **Start** menu select **Settings** → **Control Panel**.
2. Open the Network object.
3. Select the **Protocols** tab.
4. Select the TCP/IP Protocol entry and click the **Properties...** button.
5. In the Microsoft TCP/IP Properties window, select the **DNS** tab.

6. Add your domain name to the **Domain** field and optionally to the **Domain Suffix Search Order** field in the following format:  
austin.ibm.com
7. In the Microsoft TCP/IP Properties window, click the **OK** button.
8. In the Network window, click the **Close** button.
9. Restart your computer.
10. Restart the Setup.exe.

---

## Coexisting With the Hummingbird SOCK5 Server

WIN

Under particular circumstances, running Component Broker simultaneously with the Hummingbird SOCK5 server results in a General Protection Fault. This problem does not occur when the SOCK5 server is not running.

To remove the SOCK5 server, perform either one of the following steps:

- Execute the SOCK5DEL.EXE program found in the Program Files\hummingbird directory.
- Rename the HCLSOCK5.DLL file in the winnt\system32 directory.

Restart the computer. Web browsers or other programs dependent on SOCKS servers will be affected by the absence of the SOCK5 server.

To reinstate the SOCK5 server, perform either one of the following steps:

- Execute the INSTALL.BAT program found in the Program Files\hummingbird directory.
- Rename the renamed HCLSOCK5.DLL file in the winnt\system32 directory to its original name.

---

## Manually Clearing Up the Existing Configuration

WIN

If for some reason reconfiguration fails, you may want to try the following:

1. If a remote System Manager is used to manage this client system, manually delete the host object for the failed computer if it is not already deleted from the remote System Manager:

- a. Change to the drive on which Component Broker is installed:

x:

Where x: represents the Component Broker drive.

- b. Change directories to the Component Broker directory:

```
cd CBroker\data
```

Where CBroker represents the directory in which Component Broker is installed.

- c. Delete the host object:

```
bgcmd @smappl_host/HostI.client_host delete
```

Where *smappl\_host* is the host name of the System Manager host and *client\_host* is the host name of the client computer with the failed reconfiguration.

2. Return to the client computer. Save the failed configuration in the CBroker\data directory:

- a. Change to the drive on which Component Broker is installed:  
`x:`  
 Where *x*: represents the Component Broker drive.
  - b. Change directories to the Component Broker directory:  
`cd CBroker\data`  
 Where *CBroker* represents the directory in which Component Broker is installed.
  - c. Create a directory in which to save the configuration:  
`md save`  
 Where *save* represents the name of your new directory.
  - d. Save the configuration:  
`copy * save`  
 Where *save* represents the name of your new directory.
3. Stop the CBCConnector service:
    - a. From the **Start** menu, select **Settings** → **Control Panel** → **Services**.
    - b. If the CBCConnector service is not stopped, select CBCConnector and select **Stop**.
  4. Remove the failed configuration files in the *CBroke\data* directory. Ensure that any subdirectories in the *CBroke\data* directory and the files in those subdirectories are not removed:  
`del *`
  5. Copy all files from the *CBroke\data\BASE1* directory to the *CBroke\data* directory:  
`copy BASE1\*`
  6. Start the Component Broker Configuration tool. From the **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Configuration tool**.
  7. Once the configuration tool is running, select **Custom Reconfiguration**.
 

**Note:** If you have never successfully configured Component Broker, the Configure New Install window could be displayed. If this is the case, select **Custom Configuration** and click the **Next** button.
  8. Select the packages you want to reconfigure. Leave the **Overwrite existing configuration** check box selected.
  9. Continue with the reconfiguration. When a dialog window opens with the following message: “servers.txt file is missing. This could cause problems cleaning up the existing configuration. Do you want to continue with the reconfiguration?,” select the **Yes** button to continue with the reconfiguration.
  10. If the reconfiguration fails, redo step 3 through 7. In 8, leave the **Overwrite existing configuration** check box unchecked.
  11. If the reconfiguration still fails, contact your IBM representative.

---

## Unable to Start the Component Broker Configuration Tool

WIN

Make sure the JDK 1.1.6 bin directory is included in the PATH environment variable and the JDK 1.1.6 classes.zip file is included in the CLASSPATH environment variable. If multiple versions of the JDK are included in the CLASSPATH, make sure the JDK 1.1.6 occurs before any earlier version.

---

## Return Code 1069 From System Manager Configuration

WIN

During Component Broker configuration using the Component Broker configuration tool, if the following error is returned:

SM - Failed to start CBConnector service. ReturnCode = 1069.

The user ID and password you entered to start the Component Broker service is incorrect. Before running the Component Broker configuration again, bring up Services to correct the logon user information for the CBConnector service:

1. From the **Start** menu, select **Settings** → **Control Panel**.
2. Open the Services object.
3. Select CBConnector and select the **Startup** button.
4. Correct the user and password information under **Log On As This Account**.
5. Select the **OK** button.
6. Select the **Close** button to close the Services window.

Rerun the Component Broker configuration tool:

1. To start the configuration tool from the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Component Broker Configuration tool**.
2. Once the configuration tool is running, select **Custom Reconfiguration**.

**Note:** If you have never successfully configured Component Broker, the Configure New Install window could be displayed. If this is the case, select **Custom Configuration** and click the **Next** button.

3. In the Custom Reconfiguration window, select the packages you want to reconfigure. Leave the **Overwrite existing configuration** check box selected. For information about reconfiguring, see “Reconfiguring Your Component Broker Installation” on page 113.

---

## Incomplete PATH After Rebooting Component Broker or Installing Another Product

WIN

The CBConnector service is not able to start and other Component Broker applications are not able to compile or start.

Windows NT restricts the combined system PATH and user PATH to 512 bytes. Your combined system PATH and user PATH exceeded the 512 byte limit. If neither system nor user PATH by itself exceeds the 512 byte limit, then Windows NT randomly discards either system or user PATH. Otherwise Windows NT discards the portion of the PATH that exceeds the limit.

To fix this problem, follow this order of elimination:

1. Eliminate duplicate path settings found in both system and user PATH.
2. Eliminate any settings not required during reboot.
3. Reboot the computer.

If problems persist, contact your IBM representative.



---

## Database CBSRFCDB Not Created Correctly

If the CBSRFCDB database was not created correctly, perform the following steps:

1. If not already started, start db2:

```
db2start
```

2. Drop the database manually:

```
db2cmd db2 drop database CBSRFCDB
```

3. Run the script to create the database:

**WIN** For Windows NT

```
db2cmd db2 -f CBroker\etc\inscrtdb.db2
```

Where *CBroker* represents the directory in which Component Broker is installed.

**AIX** For AIX

```
db2cmd db2 -f /usr/lpp/CBConnector/etc/inscrtdb.db2
```



---

## Chapter 14. Testing the Component Broker Installation

After completing the System Manager or the Application Server installation, you can use the Component Broker System Manager to configure Component Broker Name Servers and Application Servers, and to manage Component Broker clients. This chapter provides procedures on how to configure and set up your initial Component Broker environment with the Programming Model Policy Sample Application and how to use the System Manager User Interface.

The System Manager is started when the host is started. All the administrator needs to do is start the System Manager User Interface. The System Manager User Interface can be stopped and restarted any number of times, independently of the System Management functions.

**WIN** To start the System Manager User Interface from the **Start** menu, select **Programs** → **IBM Component Broker** → **System Manager User Interface**.

**AIX** To start the System Manager User Interface from an AIX window using the cbtest user ID created during DB2 configuration, enter:

```
bhgeui '$eui' hostname
```

where *hostname* is the fully-qualified name of the host where the System Manager or the System Management Agent is installed.

This chapter contains the following sections:

- “The Initial Configuration”
- “Adding Objects to a Server” on page 150
- “Disabling Security Service” on page 151

For complete information about using the System Manager, see the *Component Broker for Windows NT and AIX System Administration Guide* .

---

### The Initial Configuration

This section explains the steps to configure and start your first Component Broker server on a host. Specific system management actions (for example, activate) cause progress windows to be displayed. You can monitor the progress of these asynchronous actions using these windows.

Configure and start a component server on a host by:

1. “The Policy Sample Application”
2. “Configuring the Policy Sample Application” on page 146
3. “Executing the Policy Sample Application” on page 148

### The Policy Sample Application

The Policy Application is provided to verify the correct operation of the Component Broker System Manager and to help guide you when creating your own applications. The Policy application consists of multiple parts:

- Client applications
- Business objects
- A home

The business objects and the home must be installed on a server and defined to System Management before the client applications can run. The Policy Application is built and ready for execution.

The PolicyApp client program demonstrates the creation of Transient Business Objects. To run the PolicyApp client program, the PolicyFamily Application Family must be properly configured to System Management.

**Note:** To compile the Policy Application, you must install the Server SDK, Application Development Tools, and the Toolkit samples from the Component Broker Toolkit CD-ROM. For instructions on how to compile the Policy Application, view the following HTML files using your web browser:

WIN

x:\CBroker\samples\IVP\ProgrammingModel\Docs\progmodel.html

where x:\CBroker represents the installation directory for Component Broker.

AIX

/usr/lpp/CBToolkit/samples/IVP/ProgrammingModel/Docs/progmodel.html

The pre-built Policy Application is not enabled for Object Level Trace. If you would like to use the Object Level Trace functionality, follow the instructions in the “OLT Scenarios” topic of the *Component Broker Application Development Tools Guide*.

## Configuring the Policy Sample Application

Perform the following steps:

1. “Define the PolicyFamily Application to System Management”
2. “Configure an Application Server” on page 147
3. “Load the Application on a Server” on page 147

**Note:** This document assumes that the sample application will be added to a server that is contained in a server group.

### Define the PolicyFamily Application to System Management

The Component Broker uses its own application installation process to add Component Broker functions to a server.

To load the application definition to your Host Images:

1. From the System Manager User Interface menu, select **View** → **User Level** → **Expert**.
2. From the System Manager User Interface Home view, expand the Host Images.
3. Select a host image. The host image corresponds to the name of the system on which the Server with System Manager configuration was installed.
4. Open the pop-up menu, and select **Load Application**.
5. Browse and select PolicyFamily.ddl

WIN

x:\CBroker\samples\IVP\ProgrammingModel\PolicyFamily\PolicyFamily.ddl

where x:\CBroker represents the installation directory for Component Broker.

AIX

```
/usr/lpp/CBConnector/samples/IVP/ProgrammingModel/Policy
/PolicyFamily/PolicyFamily.ddl
```

The Action Console window is displayed that contains a message indicating successful completion.

6. Close the Action Console window.

## Configure an Application Server

To configure an application server you need to create a *server group*. A server group is an abstraction or a template to which you add servers. A *server* in a server group inherits its attributes and applications from its server group. To configure an application server:

1. Create a Server Group in the Sample Configuration.
  - a. From the System Manager User Interface Home view, expand Management Zones → Sample Cell and Work Group Zone → Configurations, and select Sample Configuration.
  - b. Open the pop-up menu of Sample Configuration, and select **New** → **Server Group**. A dialog is displayed.
  - c. Type a Server Group name. In this exercise, *myservergroup* is used as the Server Group name.
  - d. Click the **OK** button.
2. Create a Server in your Server Group.
  - a. From the System Manager User Interface Home view, expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configurations → Server Groups, and select *myservergroup*.
  - b. Open the pop-up menu of *myservergroup*, and select **New** → **Server (member of group)**.
  - c. Type a Server name. In this exercise, “*myserver*” is used as the Server name.
  - d. Click the **OK** button.
3. Drag *myserver* to the clipboard.
  - a. From the System Manager User Interface, expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configurations → Server Groups → *myservergroup* → Servers (member of group), and select *myserver*.
  - b. Open the pop-up menu of *myserver*, and select **Drag**.
  - c. Collapse Management Zones.
4. Expand Hosts.
5. Select the host with the same name as your Host Image (in this exercise *myhost.austin.ibm.com*).
6. Open the pop-up menu of *myhost.austin.ibm.com*, and select **Configure Server (member of group)**.

Optionally, you can navigate into the hosts by expanding Hosts. Expanding the host on which you just dropped the server and expanding Configured Servers (member of group), you will see the Server you created.

## Load the Application on a Server

You now have a server assigned to a host and you informed System Management about the new application. You now need to load the Policy application on a server. To load the Policy application on a Component Broker server:

1. From the System Manager User Interface Home view, expand Available Applications, and select Policy.

2. Open the pop-up menu of Policy, and select **Drag** to move the application to the System Manager clipboard.
3. Expand Management Zones → Sample Cell and Work Group Zone → Configurations, and select Sample Configuration.
4. Open the pop-up menu of Sample Configuration, and select **Add Application**. This places the application in the configuration.
5. Expand Sample Configuration → Applications. You should see that the Applications folder is populated with the Policy application
6. Up until now you have made the application part of a configuration, but have not specified which Component Broker servers in the configuration will execute the application. To do this for the Policy application in the Applications folder:
  - a. Select Policy.
  - b. Open the pop-up menu of Policy, and select **Drag**.
  - c. Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → Server Groups and select *myservergroup*.
  - d. Open the pop-up menu of *myservergroup*, and select **Configure Application**. A Configured Applications folder is displayed. If you expand it you will see the Policy application.
7. Enable security services (optional). For information about enabling the security services, see Appendix E, “Enabling the Security Service” on page 191.

## Executing the Policy Sample Application

Now that you have a configuration, a Server Group with an application, and an application server, you can activate your configuration and run the Sample Application.

### Activating Your Configuration

To activate your configuration:

1. From the System Manager User Interface Home view, expand Management Zones → Sample Cell and Workgroup Zone → Configuration, and select Sample Configuration.
2. Open the pop-up menu of Sample Configuration, and select **Activate**.
3. Monitor the Action Console window for a completion status.
4. Close the Action Console window.

Optionally, you can use DCE to examine the contents of the Component Broker servers. For details, see “Verifying Services With DCE” on page 149.

**Note:** If you rename a server in a configuration and activate that configuration, the Host Images folder contains the new and the old server images. Servers that are no longer part of the configuration are shut down. If you want to remove the old server name, explicitly do so in the server image.

### Running the Sample Application

Now you are ready to run the Policy Sample Application.

WIN Follow the steps below to start the application:

1. At a command prompt, change directory to:

```
x:\CBroker\samples\IVP\ProgrammingModel\Policy
```

where x:\CBroker represents the installation directory for Component Broker.

2. Enter:

```
PolicyApp
```

**AIX** Follow the steps below to start the application:

1. At a command prompt, change directory to:

```
/usr/lpp/CBConnector/samples/IVP/samples/ProgrammingModel/Policy
```

2. Enter:

```
PolicyApp
```

These applications use the name service and factory finder to locate the Homes for business objects and create new objects that the applications then locate and verify.

---

## Verifying Services With DCE

To verify that the services are enabled, ensure that the services are added to DCE correctly by checking the DCE Director Cell Directory Service (CDS) domain.

**WIN** To check the DCE CDS domain:

1. From a command prompt, enter:

```
dcemapp
```

**Note:** For IBM DCE 2.0, the DCE Director was started using the `director` command.

2. Double click the CDS objects in the Cells window to look in the CDS.

3. Verify that the name server, once configured and registered, is properly stored in the DCE-CDS. From the CDS window, search through the path by double-clicking each item until you find: CBC-local-roots → (a long number) → host → resources → servers. The name server (*myhost.austin.ibm.com-Name-Server*) should be under servers in the tree view.

**AIX** To check the DCE CDS domain:

1. From a command prompt, enter:

```
cdsbrowser
```

2. Double-click:

```
/.:
```

3. Verify that the name server, once configured and registered, is properly stored in the DCE-CDS. From the CDS window, search through the path by double-clicking each item until you find: CBC-local-roots → (a long number) → host → resources → servers. The name server (*myhost.austin.ibm.com-Name-Server*) should be under servers in the tree view.

**Note:** If you do not find the name server listed within the DCE-CDS, ensure that the DCE application is correctly configured and is running.

System management uses *myhost.austin.ibm.com-Name-Server* as the default DCE principal for the name server. If *myhost.austin.ibm.com-Name-Server* already exists in the DCE Registry, system management assigns a new DCE principal for the name server. The new DCE principal for the name server is recorded in the Security Name attribute of the Security Service Image of the name server.

To find the **Security Name** attribute:

1. From the System Manager User Interface menu, become an Expert user. (**View** → **User Level** → **Expert**).
2. From the System Manager User Interface Home view, expand Host Images → *myhost.austin.ibm.com* → Server Images, and select *myhost.austin.ibm.com-Name-Server*.
3. Open the pop-up menu of *myhost.austin.ibm.com-Name-Server*, and select **Edit**. This opens the Object Editor.
4. Click the **Security Service** tab.

The DCE principal of the name server is in the **Security Name** attribute. The name server is registered with the DCE Registry with the DCE principal shown in the **Security Name** attribute.

---

## Reloading an Application

If you regenerated and recompiled your application from Object Builder after making a change, you must replace the DDL file and the executables that your Component Broker system is using.

**Note:** If there are two related Application Families due to your application using DB2 (for example, MyFamily and MySpecificFamily), the basic family must be refreshed before the specific family.

To replace these files:

1. From the System Manager User Interface menu, select **View** → **User Level** → **Expert**.
2. Expand Host Images → *myhost.austin.ibm.com* → Application Family Installs, and select *PolicyFamily*.
3. Open the pop-up menu of *PolicyFamily*, and select **Refresh**. A dialog is displayed.
4. Ensure that the dialog contains the name of the DDL file that you just generated with Object Builder. This name should be the same as the original DDL file, and the dialog should already contain this value.

---

## Adding Objects to a Server

After the sample application is installed and system management objects and application code are created, an application can be added to any running server.

1. Install the application. These steps should include:
  - Copy the Sample Application software to the host computer.
  - Create System Management objects (from the DDL file).
  - Create an Available Application System management object to represent the installed application.
2. Define the application. Drag the Application Install object and drop it onto a Configuration. You can drag an object in the Available Applications folder of the Home view or on the corresponding Application Install object in the Application Family Install object under the Host Image.
3. Configure the Application on the Server. Drag the Application Model and drop it onto the Server Model (freestanding) or Server Group Model.
4. Activate the Configuration.



---

## Disabling Security Service

This section contains procedures for disabling Security Service for the Name Server, any application server, and any client style images.

### Important

You need to stop your servers before changing your security setting to turn security either on or off. See Appendix J, “Stopping Servers” on page 215 for details.

To disable the Security Service:

1. Disable security for the Name Server.
  - a. Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → Name Server, and select Sample Name Server.
  - b. Open the pop-up menu of Sample Name Server, and select **Edit**.
  - c. Click the **Security Service** tab.
  - d. Change the **security enabled** field from yes to no.
  - e. Click the **OK** button.
2. For each server group within Management Zones under Server Groups:
  - a. Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → Server Groups.
  - b. Select *any server group*.
  - c. Open the pop-up menu of *any server group*, and select **Edit**
  - d. Click the **Security Service** tab.
  - e. Change the **security enabled** field from yes to no.
  - f. Click the **OK** button.
3. For each server (free standing) within Management Zones under Sample Configuration:
  - a. Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → Server (free standing).
  - b. Select *any server*.
  - c. Open the pop-up menu of *any server*, and select **Edit**
  - d. Click the **Security Service** tab.
  - e. Change the **security enabled** field from yes to no.
  - f. Click the **OK** button.
4. For each client style within Management Zones under Client Styles:
  - a. Select *any client style*.
  - b. Open the pop-up menu of *any client style*, and select **Edit**. This opens the Object Editor.
  - c. Click the **Security Service** tab.
  - d. Change the **security enabled** field from yes to no.
  - e. Click the **OK** button.

You have now successfully validated the run time installation. You are now ready to install additional applications. If this is your first time using Component Broker, building the claim sample is the next logical step in familiarizing yourself with building business objects and with Component Broker. For information

about the claim sample, see the “Build a Wizard” topic in the *Component Broker Application Development Tools Guide*.

---

## Chapter 15. Information About Component Broker Samples

Several sample applications are provided with Component Broker. These samples help to demonstrate the many powerful features that Component Broker has to offer. This section provides information to help you locate and use the supplied samples.

---

### General Information

The samples are located in the following paths:

**WIN** `CBroker\samples`

**AIX** `/usr/lpp/CBToolkit/samples`

Where *CBroker* represents Component Broker installation directory on the CBToolkit compact disc.

The samples are separated into the following categories:

- “Installation Verification Programs”
- “Tutorial Samples”
- “Early Test Function” on page 154

### Installation Verification Programs

The installation verification samples provide a way to verify that the Component Broker installation was done correctly. The samples consist of:

**Programming Model** Files used in examples documented in the *Component Broker for Windows NT and AIX Planning, Performance, and Installation Guide* and in the *Component Broker for Windows NT and AIX Quick Beginnings*.

**PAA** Files used in examples documented in the *Component Broker for Windows NT and AIX CICS and IMS Application Adaptor Quick Beginnings*.

**Oracle** Files used in examples documented in the *Component Broker for Windows NT and AIX Oracle Application Adaptor*.

The installation verification samples are located in the following directories:

**WIN** `CBroker\samples\InstallVerification`

**AIX** `/usr/lpp/CBToolkit/samples/InstallVerification`

Where *CBroker* represents Component Broker installation directory on the CBToolkit compact disc.

### Tutorial Samples

The tutorial samples demonstrate Component Broker programming model concepts and patterns. The samples consist of:

**Composite Business Objects** Samples of two styles of composite objects.

**Fundamentals** Samples that guide you through the fundamental concepts of the Component Broker programming model.

**Personal Life Insurance** Sample that demonstrates Team Development using the application example in Chapter 2 of the *Component Broker Programming Guide*.

**QuickTest** This 100% Java Application consists of an Emitter and a Framework. The Emitter processes the IDL files generated from Object Builder and emits GUI Java classes that access the Business Objects on the server. The QuickTest Framework works with the emitted classes and provides a complete GUI Java Application/Applet that allows testing of the server application.

**Workload Management** Sample demonstrating a Workload enabled application.

The tutorial samples are located in:

 **Windows NT** *CBroker*\samples\Tutorial

 **AIX** /usr/lpp/CBToolkit/samples/Tutorial

Where *CBroker* represents Component Broker installation directory on the CBToolkit compact disc.

## Early Test Function

The early test function samples demonstrate Component Broker functions that are in the early test phase. These samples consist of:

**Enterprise Java Beans** Demonstrates the initial Enterprise JavaBean support in Component Broker.

**Business Rules** Contains the information and code for the Accessible Business Rules Framework.

The early test function samples are located in:

 **Windows NT** *CBroker*\samples\EarlyTestFunction


 **AIX** /usr/lpp/CBToolkit/samples/EarlyTestFunction


Where *CBroker* represents Component Broker installation directory on the CBToolkit compact disc.

## Documentation for the Samples

There is additional documentation for the samples available for you to use. The Samples.html file includes links to most of the sample documentation. In addition, the CICS and IMS application adapter and Oracle application adaptor samples have their own Quick Beginnings guides containing specific documentation.

The sample documentation can be viewed as follows:

 **Windows NT** From the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Sample Information**.

 **AIX** Use your web browser to view /usr/lpp/CBToolkit/samples/Samples.html

---

## Generating the Business Objects for the Samples

The samples in the Install Verification and Tutorial categories are shipped with Object Builder model files instead of actual source code. You can use these files to generate the code for the business objects.

### Notes:

1. To utilize the samples on AIX you need to copy them from the Component Broker installation directory located under `/usr/lpp/CBToolkit/samples` into your user space. From there you can generate the code and compile it. To copy the files into your user space, enter the following command:

```
cp -r /usr/lpp/CBToolkit/samples $HOME
```

2. To accommodate the generated sample code for all of the samples, 250 megabytes of disk space is required.

To generate the code for all of the samples, you must first change to the directory containing the samples. In Windows NT, enter the following command:

```
cd CBroker\samples
```

Where *CBroker* represents the directory in which the samples are located.

In AIX, enter the following command:

```
cd $HOME/samples
```

Generate the code by running the following command:

```
gensrc platform
```

Where *platform* represents one of the following:

- NT
- AIX
- 390

### Notes:

1. If you do not specify the platform, the default is the platform on which you are currently running.
2. The gensrc command is located in the following directories:

 **Windows NT** *CBroker\samples*

 **AIX** `/usr/lpp/CBToolkit/samples`

Where *CBroker* represents Component Broker installation directory on the CBToolkit compact disc.

To generate source for only the samples in which you are interested, you can run the gensrc commands located in the following directories:

```
...\samples\Tutorial
...\samples\Tutorial\Fundamentals
...\samples\Tutorial\Insurance
...\samples\Tutorial\CBOB
...\samples\Tutorial\WLM
...\samples\InstallVerification
...\samples\InstallVerification\OAA
```

...\samples\InstallVerification\PAA

...\samples\InstallVerification\ProgrammingModel

---

## Compiling and Running the Samples

To compile and run the samples, perform the following steps:

1. Compile the business objects. You can find information to help you compile the business objects for each of the samples in each sample's documentation.

**Note:** The sample documentation can be viewed as follows:

**WIN** **Windows NT** From the Windows NT **Start** menu, select **Programs** → **IBM Component Broker** → **Sample Information**.

**AIX** **AIX** Use your web browser to view `/usr/lpp/CBToolkit/samples/Samples.html`

2. Load each of the sample servers. You can find information to help you load the sample servers in "Configuring the Policy Sample Application" on page 146. You need to use the .ddl provided by each sample.
3. Run the clients. You can find information to help you run the clients in the sample's documentation.

---

## Programming Model and JDK Code Examples

The QuickTest utility provides the source code for its framework. To see Component Broker Java Client Programming Model examples used within the QuickTest framework, browse the following page:

**WIN** **Windows NT** `CBroker\samples\QuickTest\framework\QThowToPM.html`

**AIX** **AIX** `/usr/lpp/CBToolkit/samples/QuickTest/framework/QThowToPM.html`

Where *CBroker* represents Component Broker installation directory on the CBToolkit compact disc.

To see Java and Java Foundation Class examples used within the QuickTest framework, browse the following page:

**WIN** **Windows NT** `CBroker\samples\QuickTest\framework\QThowToJDK.html`

**AIX** **AIX** `/usr/lpp/CBToolkit/samples/QuickTest/framework/QThowToJDK.html`

Where *CBroker* represents Component Broker installation directory on the CBToolkit compact disc.







---

## Chapter 16. Performance Monitoring and Tuning

The following topics in this section provide information on how you can monitor and tune Component Broker application performance during run time:

- “CBCConnector Tuning”
- “Operating System Tuning” on page 160
- “Java Virtual Machine Tuning” on page 161
- “Database Tuning” on page 163
- “Performance Monitoring” on page 163

In addition, the following references provide information on enhancing Component Broker application performance:

- Chapter 2, “Configuration and Capacity Planning” on page 9
- The “Performance Considerations” topic in the *Component Broker Advanced Programming Guide*.

---

### CBCConnector Tuning

Some CBCConnector components offer tuning attributes to help achieve better application performance. Many of these attributes are accessible using the System Management EUI.

**Note:** See the *Component Broker for Windows NT and AIX System Administration Guide* for information about using the System Management EUI.

- The System Management Agent process (bgmain) runs periodically to check CBCConnector server status and to insure that one is always active. This polling activity can consume valuable CPU cycles. To minimize the contention for the CPU between the agent and the CBCConnector servers the polling frequency can be adjusted or eliminated completely.

Edit the `Host Image` and modify the `health polling` attribute. Setting the value to 0 turns off polling completely.

If health polling is turned off it is recommended that the `Run On Request` attribute be enabled. This forces a CBCConnector server to start automatically in the event one is not active.

- A certain amount of baggage is carried along with each client initiated request. One of these pieces of baggage is used to log client specific information on the server. This information is useful when diagnosing client and server problems.

If performance requirements supercede the need for “first failure data capture,” the `RAS request` and `response filters` can be turned off. Doing so significantly increases client response time and server throughput.

Edit the `Client Image` and `Server Image`. Set the `RAS filters` attribute in the `Log Controls` window to `no`.

- For environments where sessions are not used, the `sessions enabled` attribute in the `Client Image` can be turned off. This helps eliminate some unnecessary processing on client systems that do not use sessions.
- Multi-processor systems provide opportunities as well as challenges for multi-threaded applications. The amount of resource sharing between threads is a key factor in how well multi-threaded applications perform on Symmetric Multi-Processor (SMP) systems.

To achieve better server throughput, it is recommended that you dedicate the server to a single processor to minimize some of the resource contention that occurs on SMP systems.

Each operating system has its own mechanism for doing this. On Windows NT you can use the TaskManager. Right click the somsrsm process to display the **Set Process Affinity** menu. To dedicate a CBCConnector server process (somsrsm), select only one processor. This must be done each time you start a server. On AIX you can use the bindprocessor command. See the Manual page or InfoExplorer for details on the command syntax.

- CBCConnector server requests are handled by individual threads within the server. As method requests arrive at the server they are read from a socket on one of the ORB threads and dispatched on separate server run-time threads. Each of these threads are maintained in separate “thread pools.”

The number of simultaneous inbound requests that can be handled efficiently by both of these thread pools is directly related to how many of these threads are available to run. If these thread pools are exhausted or need to be expanded, delays in processing time can occur.

The initial number of threads available in each of these pools should be adjusted based on the peak loads the server is expected to handle. For example, if the server is expected to deal with 100 clients in a peak load scenario it is recommended that these two thread pools be at least 50. This will minimize the amount of time the server has to wait for free threads to become available.

To adjust the default thread pool values use the System management EUI and edit the Server Image of interest (Application or Name server). To adjust the ORB thread pool click on the **ORB** tab; to adjust the server run time thread pool click on the **Main** tab.

- The CBCConnector transaction service uses a hard disk file (transaction log) to maintain a history of transactions in progress. This log allows the transaction service to maintain database integrity in the event of a transaction or system failure. Because of this dependency on the hard disk, CBCConnector Application Server performance is dependant on the speed of the disk on which the transaction log is located.

To achieve optimum transaction performance the transaction log should be located on the fastest disk possible and preferably one that is independent of operating system paging activity as well as database usage. Simple benchmarks are available to test your disk performance. Manufacturer's specifications are also a good source of information about your current hard disk performance capabilities. Disk rates that perform in the ten to twenty megabyte per second range are preferred.

The default location for the transaction log is the drive where CBCConnector is installed. To change this location you can use the Systems Management EUI and edit the Server Image by selecting the Application server of interest and clicking on the **Transaction Service** tab to change the default location of the transaction log.

---

## Operating System Tuning

This section points out Operating System specific functions that could be tuneable and that could help CBCConnector performance. Many more details regarding operating system tuning can be found in the following references.

For the Windows NT operating system, the Windows NT Resource Kit contains a wealth of information regarding performance monitoring, diagnosing performance problems, and available performance tools. For information, see the *Microsoft Windows NT Workstation Resource Kit*.

A good source of information for the AIX platform is the *AIX Performance Tuning Guide Versions 3.2 and 4*, SC23-2365. This book includes several discussions on various general performance topics as well as AIX-specific performance topics.

- One of the more important operating system parameters that effects application performance is the paging space configuration. The paging space located on physical disk storage is used by the systems Virtual Memory Manager to resolve virtual memory addresses. Most operating systems recommend

that the size of the paging space be at least two times the systems physical memory size. That is, if your physical memory is 128 megabytes your paging space size should be at least 256 megabytes. See your operating system manual for specific recommendations.

An additional consideration regarding the paging space for your system is the physical location of the paging file (that is, which physical disk). For optimum performance you should locate your paging space on a disk that is separate from where the CBCConnector transaction log and Database are located. The default location of the CBCConnector transaction log is the drive where CBCConnector was installed. The CBCConnector transaction log may be relocated using the Systems Management EUI.

This tip helps to minimize disk contention between the operating system, CBCConnector, and Database managers.

- Process priority is another key attribute that contributes to a process's ability to get work done in a timely fashion. The only time to consider adjusting a CBCConnector process priority is when that system is not being used as a dedicated CBCConnector Server. If the system has multiple roles (for example, CBCConnector Server, Print Server, Mail Server, and so on), it could be appropriate to adjust the CBCConnector Server process priority to be more favored than the other non-critical tasks. Each operating system has a different means to accomplish this task. For example, on Windows NT the TaskManager can be used and on AIX the `setpri` command could be appropriate. See the operating system documentation for details.

---

## Java Virtual Machine Tuning

Before running JavaBOs or CICS/IMS, certain attributes for the Java Virtual Machine can be customized from System Management. You can usually use the defaults for these attributes. Occasionally these attributes must be changed when an application is deployed in a different environment. For example, as you increase the number of objects being used, the Java virtual machine eventually runs out of memory. In this case, the heap size for the virtual machine must be increased.

Perform the following steps to change the Java virtual machine attributes:

1. In System Manager, from the **View** menu, select **User Level** → **Expert**. This causes the System Manager window to display more detailed information and also allows you to change the settings for the Java virtual machine.
2. Open the Host Images folder, and expand the host image that corresponds to the name of the server.
3. Expand Server Images and locate the server image in which the applications resides.
4. From the pop-up menu for the server image, select **Edit** to open the notebook.
5. Select the **Java Virtual Machine** tab.
6. Change the settings of each attribute as required. The customizable attributes are:
  - native stack size** Size of the native stack in megabytes for each Java thread.
  - java stack size** Size of the Java stack in megabytes for each Java thread.
  - minimum heap size** Minimum heap size in megabytes for storing Java objects. For example, 32 means 32 megabytes.
  - maximum heap size** Maximum heap size in megabytes for storing Java objects. This setting should be increased when the `java.lang.OutOfMemory` error is detected.
  - enable class garbage collection** Set to a non-zero value to enable garbage collection of Java class objects.
  - enable verbose garbage collection** Set to a non-zero value to tell the Java virtual machine to print messages on each occurrence of garbage collection.

**properties** Properties to be added to the virtual machine's system properties in the format `key=value`. You can specify multiple properties using `#` as the separator. For example:

```
java.compiler=myjit#awt.button.color=green
```

defines the following two properties:

- `java.compiler=myjit`
- `awt.button.color=green`

**log VM activity** Log Java VM activity to the console. This primarily reports which Java classes are loaded and from where.

The minimum and maximum heap size parameters for the Java virtual machine can be set using system management. The Java virtual machine uses the minimum heap size parameter to allocate the initial storage for Java objects. When memory allocation exceeds this initial size, the Java virtual machine first tries garbage collection. Only when the garbage collection fails to free sufficient memory does the virtual machine allocate additional heap space. The maximum amount of heap space that the virtual machine allocates is determined by the maximum heap size parameter. If during an allocation the maximum heap size is exhausted, and the garbage collector fails to free sufficient memory, a `java.lang.OutOfMemory` error is raised.

Care must be taken when tuning the minimum heap size. JavaBOs interact with objects on the C++ server using Java proxies to these objects. These proxies and the corresponding C++ implementation objects to which they point are not freed until the proxies are garbage collected. Since C++ objects can be an order of magnitude bigger than their lighter-weight Java proxies, setting the initial heap size too large can cause over-commitment of C++ memory on the server. For example, setting the minimum heap size to 32 megabytes could cause C++ memory allocations of more than 320 megabytes even before the first garbage collection starts to run. This problem can be avoided by setting the minimum heap size smaller. Garbage collection then runs more frequently and dramatically reduces the C++ memory requirements.

For the above reasons, the minimum heap size should only be modified when the memory requirement of an application is known. For example, the application may pull in additional Java class libraries whose run-time sizes are known. Or experimentation may have determined the memory usage requirements. In these cases, enlarging the minimum heap size can potentially result in better performance due to fewer garbage collections before sufficient memory is allocated for the application's working set.

The following steps can be taken to diagnose JavaBO memory related problems for an application:

1. If the activity log reports a `java.lang.OutOfMemory` error (or `UNKNOWN ERROR` if the server could not log the `OutOfMemory` exception correctly because of the out-of-memory status), try increasing the maximum heap size. If increasing the maximum heap size results in ever increasing memory allocation, the potential causes are:
  - The real memory requirement for the application is too large for the machine on which it is run. Try migrating the application to a machine with larger memory and more disk space.
  - There is a memory leak on the Java side. This can happen if a programming error causes an application to allocate more memory to a data structure without removing the references to existing structures. In this case, Java garbage collection will not be able to collect any object.
  - There is a memory leak on the C++ side. This can happen if there are interactions between JavaBOs and C++ BOs, and there is a leak in C++ BOs that keeps portions of JavaBOs in memory and inaccessible to garbage collection.
  - There is a problem within Component Broker related to the interaction between Java and C++ objects. Contact your Component Broker representative for additional help.

2. If the activity log does not report a `java.lang.OutOfMemory` error (or `UNKNOWN ERROR`), try decreasing the minimum heap size. This should force garbage collection to occur more frequently, reducing the load on C++ memory requirements. To observe whether or not garbage collection runs, turn on the `enable verbose garbage collection` parameter for the Java VM settings from system management. If lowering the minimum heap size does not solve the problem, the potential causes are:
  - The C++ memory requirements for the application are too large for the machine on which it is running. Try migrating the application to a machine with larger memory and more disk space.
  - If C++ BOs are also installed on the same server, there is a memory leak originating from the C++ BOs.
  - There is a problem within Component Broker related to the interaction between Java and C++ objects or related to just C++ objects. Contact your Component Broker representative for additional help.

---

## Database Tuning

Each database has its own set of recommendations for tuning. Rather than repeat them here the following references are good sources of information regarding specific database tuning options.

For DB2, see the “Tuning Application Performance” (Part 3) and the “Tuning and Configuring Your System” (Part 4) topics in the *Administrative Guide* in the DB2 Information Center.

---

## Performance Monitoring

Each operating system has its own set of tools that allow you to monitor system performance. The intent of this section is not to get into a detailed discussion of how to use these tools, but rather to point you in the direction of some reference material and common techniques used to look at CBCconnector performance.

For the Windows NT operating system, the Windows NT Resource Kit contains a wealth of information regarding performance monitoring, diagnosing performance problems, and available performance tools. For information, see *Microsoft Windows NT Workstation Resource Kit*.

For the AIX platform, a good source of information is the *AIX Performance Tuning Guide Versions 3.2 and 4*, SC23-2365. This book includes several discussions about various general performance topics as well as AIX-specific performance topics.

The DB2 documentation is a good source for learning how to monitor DB2 performance and includes tips on setting up a general relational database for efficiency. The *IBM DB2 Universal Database for Windows NT Administration Getting Started Version 5*, S10J-8154 will help with some basic DB2 performance tutorials. Another DB2 resource is the Performance Configuration SmartGuide. From within the DB2 Control Center, right click on the database you want to configure for performance to display the pop-up menu, select **Configure Performance** and follow the bouncing ball.

Five key performance areas to monitor when looking at CBCconnector and Application performance are:

- CPU time
- Real Memory Utilization
- Disk Accesses
- LAN/Communications traffic
- Database efficiency

A detailed discussion of each of these performance attributes, the tools you can use, and what you should look for when determining the source of a performance bottleneck can be found in the previously mentioned references.

A short introduction to some available tools and how they can be used in CBCConnector situations follows. See the Windows NT or AIX documentation for details on the various options you can use with the tools or commands.

## CPU time

### NT Tools

TaskManager

### AIX Tools

iostat, vmstat, tprof, and ps commands.

### Tips

- Use the tools listed in this section to determine what the overall CPU utilization of the system. If excessively high, (that is, less than 10% idle) determine which process or processes may be using the majority of the CPU.
- Pay close attention to the following processes:
  - somsrsm (CBCConnector Application and Name Server)
  - db2syscs (DB2 daemon)
  - somorbd (ORB daemon)
  - secd (DCE security daemon)
  - dced (DCE daemon)
  - cdsd (DCE cell directory daemon)
  - bgmain (CBCConnector system management agent)

These processes pertain to CBCConnector processing. If any one of these processes runs continuously with utilization greater than 70%, you should investigate to see if there is a CPU bottleneck.

## Real Memory Utilization

**NT Tools** TaskManager

**AIX Tools** ps, vmstat, and lps commands

### Tips

- Use the tools listed in this section to check memory utilization on the system. Poor memory utilization can sometimes be caused by “leaking” applications. Pay special attention to the processes listed in the previous section. If their memory usage continually climbs, this could indicate a possible memory leak.
- The rate at which an application is page faulting is an important attribute to monitor. See the above documentation for details. Excessive page faulting can be an indicator of a memory bottleneck.
- If the available real memory is less than 30%, the system is approaching a point where excessive page faulting will start to occur due to the shortage.
- Very high memory utilization in the Application Server process (somsrsm) is sometimes caused by Managed Objects not being released properly by the client applications or having the wrong “memory management policy” configured for the Managed Object container.

## Disk Accesses

**NT Tools** NT Performance Monitor

**AIX Tools** iostat

**Tips**

- Use the tools listed in this section to check for disk or I/O wait time. If this number is excessive (that is, if I/O wait is greater than 30%) you may have a disk bottleneck.
- When using the Windows NT TaskManager to monitor the CPU, if you notice that there is a considerable amount of idle time but the system seems relatively busy (that is, the disk light continually flashes) this is usually attributed to some amount of disk time that does not show up in the CPU utilization metric.
- If you encounter excessive disk I/O activity, ensure that the following disk files are not being written continuously due to an error that is occurring:
  - NT activity log
  - AIX error log
  - CBConnector activity log
  - DB2 error log

**LAN/Communications traffic**

**NT Tools** NT Performance Monitor

**AIX Tools** netstat, netpmon

**Tips**

- Use the tools listed in this section to check inbound/outbound communications traffic rates as well as protocol and device statistics.
- Traffic rates that approach the limits of the communications line bandwidth may indicate a networking bottleneck.
- Evaluate network device (that is, token ring, ethernet, atm) statistics to determine how efficiently a particular device is running. Adapter or device driver errors usually translate into lost packets which ultimately result in application time delays.
- Evaluate protocol statistics (that is, re-transmissions or dropped packets) to determine how efficient a particular protocol is running. CBConnector uses IP, TCP, and UDP for the most part. In addition to these, SNA LU6.2 is used in some cases for PAA support.
- ORBs are essentially sophisticated socket application programs. CBConnector relies heavily on ORB functionality to achieve its distributed network characteristics. It is therefore important to have all lower level networking components running as optimally as possible.

**Database Efficiency**

**NT Tools** DB2 Event and Snapshot Monitors

**AIX Tools** None.

**Tips**

- The DB2 Event and Snapshot monitors provide an abundance of performance information regarding database behaviour. See the *IBM DB2 Universal Database for Windows NT Administration Getting Started Version 5* for specifics on how to use these tools.

- Ensure that the DB2 error log is not being written excessively. Too many DB2 errors can cause unnecessary disk I/O activity to occur due to error logging. Always check the DB2 error log for repeated errors when disk I/O rates seem unusually high.







---

## Appendix A. Sample Response File

This appendix contains a sample response file. You can use this appendix to familiarize yourself with the layout and format of the response files used by Component Broker.

```
[InstallShield Silent]
;
; Header section
;
Version=v3.00.000
File=Response File
[Application]
Name=ComponentBroker
Version=CURRENT_VERSION
Company=IBM
;
; Dialog sequence section
;
[DlgOrder]
Dlg0=SdWelcome-0
Dlg1=AskDestPath-0
Dlg2=AskOptions-0
;
; Dialog names to be used for adding components to an existing installation.
; If this set of dialogs is used, comment out Dlg1 and Dlg2 above and
; uncomment the following three dialogs. Then re-sequence the rest of the
; dialogs in this section.
;
;Dlg1=MessageBox-0
;Dlg2=AskOptions-0
;Dlg3=SdAskOptionsList-0
;
;
; If Server and Java Client packages are going to be co-existed on the same
; system, uncomment the dialog name here and replace # in Dlg# with the
; proper sequence number. Then re-sequence the rest of the dialogs in
; this section.
;
;Dlg#=AskYesNo-0
;
Dlg3=SdShowDlgEdit1-0
;
; If documentation is to be installed and document search engine has not
; already installed on this system, then uncomment the following dialog
; name and replace # in Dlg# with the proper sequence number. Then
; re-sequence the rest of the dialogs in this section.
;
;Dlg#=SdShowDlgEdit1-1
;
; If there is a potential problem with your path environment variable,
; then uncomment the following dialog name and replace # in Dlg# with
; the proper sequence number. If the dialog name AskYesNo-0 is uncommented
; above because of selection of both Server and Java Client packages,
; change the dialog name here from AskYesNo-0 to AskYesNo-1 too.
```

```

; re-sequence the rest of the dialogs in this section.
;
;Dlg#=AskYesNo-0
;
Dlg4=AskOptions-1
Dlg5=SdStartCopy-0
Dlg6=SdFinish-0
;
; Remember to adjust the dialog count accordingly depending on the
; changes you make to the sample selection.
;
Count=7

;
; Dialog data section
;
[SdWelcome-0]
Result=1
[AskDestPath-0]
;
; If Component Broker does not exist on this system, change the directory
; path where Component Broker is to be installed. The default is C:\CBROKER.
;
szPath=C:\BROKER
Result=1
[AskOptions-0]
Result=1
;
; Currently set Sel-0 to 1 for System Manager.
; If Application Server is to be installed, set Sel-1=1 instead.
; If Standalone Workstation is to be installed, set Sel-2=1 instead.
; For Custom install, set Sel-3=1 instead and uncomment the dialog
; section [SdAskOptionList-0] below with your choice of packages
; to install. Remember to add the dialog name SdAskOptionList-0 to
; the dialog sequence section above.
;
Sel-0=1
Sel-1=0
Sel-2=0
Sel-3=0
;
; If adding components to an existing installation, uncomment the following
; three dialog sections to use and remove the two dialog sections above
; (AskDestPath-0 and AskOptions-0). Also change the dialog names in the dialog
; sequence section to reflect the change here.
;
;[MessageBox-0]
;Result=1
;[AskOptions-0]
;Result=1
;Sel-0=0
;Sel-1=1
;[SdAskOptionsList-0]
;Component-type=string
; Do not uncomment this line. Pick your choice of packages to install.
; Do not uncomment this line. Adjust the count and component sequence
; Do not uncomment this line. accordingly.
;Component-count=8

```

```

;Component-0=Server
;Component-1=C++ Client
;Component-2=Active X Client
;Component-3=Java Client
;Component-4=System Manager User Interface
;Component-5=System Manager
;Component-6=System Management Agent
;Component-7=Application Adaptor
;Component-8=Documentation
;Result=1
;

;
; If your custom selection includes both Server and Java Client, uncomment
; the following dialog section to use and add the dialog name in the dialog
; sequence section above.
;
;[AskYesNo-0]
;Result=1
;

;
; If SSL library has already installed on this system, then remove the
; following dialog section and remove its name entry in the dialog
; sequence section above. Otherwise, change the directory where you
; want the SSL library to be installed. The default is C:\GSK.
;
[SdShowDlgEdit1-0]
szEdit1=C:\GSK
Result=1

;
; If documentation is selected to be installed and document search
; engine is not installed on this system, uncomment the following
; three lines dialog section and the dialog name in the dialog
; sequence section above. Change the directory where you want
; the search engine to be installed. The default is C:\IMNNQ.
;
; Documentation search engine may have installed when you installed
; DB2.
;
; System Manager and Standalone Workstation include documentation package.
;
;[SdShowDlgEdit1-1]
;szEdit1=C:\IMNNQ
;Result=1

;
; If install detects a potential problem in your path environment variable,
; a warning message is displayed to view the Late Breaking News for more
; information.
;
; Please view the Late Breaking News. Then uncomment the following
; dialog if current path environment variable is going to cause install
; to give you a warning.
; Default is to terminate installation. If you want install to continue,
; change Result=1 to Result=0. Also if you already have a dialog named
; AskYesNo-0 because of selection of both Server and Java Client to be

```

```

; installed, change the name to AskYesNo-1.
;
;[AskYesNo-0]
;Result=1

;
; If packages to be installed does not include Server or Application
; Adaptor, remove the following three lines dialog section. Remove the
; dialog AskOption-1 in the dialog sequence section above.
;
; System Manager, Application Server, and Standalone Workstation by default
; include Application Adaptor package.
;
[AskOptions-1]
Result=1
Sel-0=1
[SdStartCopy-0]
Result=1
[SdFinish-0]
Result=1
;
; If you don't want to read the readme file after reboot, set bOpt1=0 instead.
; The setting of this option is valid only if this is the first time
; you install Component Broker on the system. Otherwise, this option
; is ignored.
;
bOpt1=1
;
; If you don't want to bring up configuration tool after reboot,
; set bOpt2=0 instead. The setting of this option is valid only
; if the package(s) you select includes Server, C++ Client,
; ActiveX Client, System Manager, System Management Agent, and/or
; System Manager User Interface. Otherwise, this option is ignored.
;
bOpt2=1

```

---

## Appendix B. Advanced DB2 Concerns

This appendix provides information on installing a Component Broker server on a DB2 client system and information on accessing multiple DB2 databases.

---

### Dropping an Existing Database on a DB2 Server

If installing the Component Broker server when there is an existing CBSRFCDB database, the existing database is not dropped during the installation because it could contain user data. In addition, the database that is normally created as part of the installation fails because the database already exists.

If this is the case, you may want to drop the database in order to clean it up prior to installation. On the DB2 server system, perform the following steps to drop the database:

WIN

1. At a Windows NT command prompt, enter: `db2cmd db2`
2. At a Windows NT command prompt, enter: `drop database CBSRFCDB`

AIX

1. Logon as the DB2 user ID.
2. In a command window, enter: `db2 drop database CBSRFCDB.`

After dropping the database and extracting your data, either perform the Component Broker server installation (if not already done) or recreate the database following the instructions in "Installing a Component Broker Server on a DB2 Client."

---

### Installing a Component Broker Server on a DB2 Client

If you are installing the Component Broker server on a DB2 client system, you will receive error messages. These messages are displayed in a DB2 command window. Before performing the following procedure, close the window.

1. Ensure that the CBSRFCDB database is created on the DB2 server machine. If not, perform the following steps on the DB2 server system:

WIN

At a Windows NT command prompt, enter:

```
db2cmd db2
create database CBSRFCDB
connect to CBSRFCDB
create table RCTABLE (KEY INTEGER NOT NULL,
                     HANDLE VARCHAR (1000) FOR BIT DATA NOT NULL)
create table RCHMTABLE (KEY VARCHAR (250) FOR BIT DATA NOT NULL PRIMARY KEY,
                       TYPEID VARCHAR (500) FOR BIT DATA NOT NULL)
create table KRCTABLE (KEY INTEGER NOT NULL,
                      HANDLE VARCHAR (1000) FOR BIT DATA NOT NULL,
                      KEYDATA VARCHAR (1000) FOR BIT DATA NOT NULL)
connect reset
quit
```

AIX

In a command window, logon as the DB2 user ID, then enter:

```

db2
create database CBSRFCDB
connect to CBSRFCDB
create table RCTABLE (KEY INTEGER NOT NULL,
                    HANDLE VARCHAR (1000) FOR BIT DATA NOT NULL)
create table RCHMTABLE (KEY VARCHAR (250) FOR BIT DATA NOT NULL PRIMARY KEY,
                      TYPEID VARCHAR (500) FOR BIT DATA NOT NULL)
create table KRCTABLE (KEY INTEGER NOT NULL,
                     HANDLE VARCHAR (1000) FOR BIT DATA NOT NULL,
                     KEYDATA VARCHAR (1000) FOR BIT DATA NOT NULL)
connect reset
quit

```

**Note:** Normally, the installation procedure creates the database. If the database already exists from a previous Component Broker installation, the database may not be compatible with this version of Component Broker. If the user wants to keep the existing database because it contains user data, the user needs to migrate the database to the current version of database that Component Broker supports. Consult the *IBM DB2 Universal Database for Windows NT Quick Beginnings Version 5*, S10J-8149 or the *IBM DB2 Universal Database for UNIX Quick Beginnings Version 5* for the procedure in migrating the database. If there is no user data in the database, then you can safely remove it. For more information, see “Dropping an Existing Database on a DB2 Server” on page 173.

2. Ensure that TCP/IP is configured between the DB2 server and the DB2 client.

WIN For additional information, see “Configuring the DB2 Server for Remote Client Access” on page 43.

3. Ensure that the DB2 security server is started on both the DB2 server system and the DB2 client system.
4. On the DB2 client system, perform the following steps:

WIN At a Windows NT command prompt, enter:

```

db2cmd db2
catalog tcpip node db2node remote hostname server service_name
catalog database CBSRFCDB as CBSRFCDB at node db2node
update dbm cfg using TP_MON_NAME somtrx1i
quit

```

AIX In a command window, logon as the DB2 user ID. Run the following commands:

```

db2
catalog tcpip db2node remote hostname server service_name
catalog database CBSRFCDB as CBSRFCDB at node db2node
update dbm cfg using TP_MON_NAME libsomtrx1.so
quit

```

Where *hostname* is the hostname of the DB2 server system, *db2node* is the node name that refers to this host, and *service\_name* is the connection port name. “somtrx1i” is with the number 1.

These DB2 configurations accomplish:

- Catalogs the DB2 server node.
- Catalogs the CBSRFCDB database.

5. Bind Reference Collection (.bnd) files to the CBSRFCDB database:

WIN



```

db2cmd db2
connect to CBSRFCDB user db2userid using password
bind x:\CBroker\etc\DBConnector.bnd
bind x:\CBroker\etc\KeyedDBConnector.bnd
bind x:\CBroker\etc\ICollectionsBOIMDOImpl_I.bnd
connect reset
quit

```

**AIX**

```

db2
connect to CBSRFCDB user db2userid using password
bind /usr/lpp/CBConnector/etc/DBConnector.bnd
bind /usr/lpp/CBConnector/etc/KeyedDBConnector.bnd
bind /usr/lpp/CBConnector/etc/ICollectionsBOIMDOImpl_I.bnd
connect reset
quit

```

Where *db2userid* and *password* are the ID and password for accessing remote DB2 server and /usr/lpp/CBConnector/etc is the base directory where Component Broker is installed.

---

## Advanced Access to DB2

**WIN** It is necessary for the DB2 Security Server to be started for the connections to be made correctly. The DB2 Security Server should automatically be started with the logon as the System account. You can verify or modify the startup values for the DB2 Security Server by modifying this service from the Services folder (Settings → Control Panel).

This information is required only for advanced scenarios where a Component Broker server is accessing multiple DB2 databases and tables.

To modify your associated server images with the appropriate user ID and password, perform the following procedure.

1. From the System Manager User Interface home view, expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → RDB Connections and select a specific database.
2. Open the pop-up menu for the selected database, and select **Edit**. This opens the Object Editor.
3. Select the **Main** tab.
4. Edit the openString attribute box. This field contains the name of your database. Modify this field to include the user ID and password for accessing this database. If your database is ClaimDB, modify:
 

```

ClaimDB
to:
ClaimDB, userid, password

```
5. Click the **OK** button to validate and save these changes.

Repeat this procedure for each RDB Connection in the RDB Connection Images folder.

After modifying all of your RDB Connections, you must reactivate your configuration to populate these changes into your host and server images.

---

## Configuring DB2 For Use with Component Broker

This section includes the following topics:

- “Next Key” on page 176
- “Settings of Timeout Values” on page 176

### Next Key

DB2 Universal Database Version 5.0 (UDB) includes a feature called Next Key Locking, which is required to guarantee ANSI and SQL 92 standard repeatable read (RR) isolation level. The Component Broker programming model is not consistent with the Next Key Locking feature of UDB enabled: the Component Broker programming model does not specify any restrictions with respect to multiple clients creating objects. However UDB's Next Key Locking feature could cause multiple clients to deadlock when multiple clients are creating objects of the same type.

Next Key Locking can be disabled for an individual UDB database by setting the `DB2_RR_T0_RS` flag to YES for any instance containing the affected databases. This disables Next Key Locking for all applications accessing all databases within the specified instances. DB2 recommends using a single database per DB2 instance to minimize the impact on other applications.

By default, UDB's Next Key Locking is enabled to guarantee the ANSI/SQL 92 standard RR isolation level. This is attained by automatically locking the next key for all INSERT and DELETE statements for all isolation levels and the next higher key value above the result set for SELECT statements for RR isolation level only. The Next Key Locking done during the INSERT and DELETE statements is required regardless of current isolation level to ensure that ANSI/SQL 92 standard RR behavior is possible for SELECT statements that can be executed by other transactions.

If any applications that share these databases with Component Broker require the ANSI/SQL 92 standard RR isolation, setting the `DB2_RR_T0_RS` flag may not be a viable solution to this problem. In that case, you must modify your Component Broker applications to handle a potential next key deadlock condition.

Other means of modifying your Component Broker application to minimize the possibility of and the impact of a next key deadlock is to use atomic transactions when possible and when applicable, or to minimize the scope of the transaction so that locks in the database are not held longer than is absolutely necessary.

### Settings of Timeout Values

From a Component Broker application programming standpoint, symptoms of a potential deadlock or race condition include your Component Broker application appearing to hang either indefinitely or for a period of time before catching one of the following exceptions:

**CORBA::NO\_RESPONSE** Indicates that no response was received for an ORB request within a specified time limit (an ORB request timeout).

**CORBA::PERSIST\_STORE** Indicates that a problem was encountered attempting to access a persistent storage device (that is, a DB2 database) or creating, retrieving, updating, or deleting a record in a persistent store.

**CORBA::TRANSACTION\_ROLLEDBACK** Indicates that during the processing of a transaction a problem was encountered requiring the transaction to be aborted and automatically rolled back.

These symptoms may also be indicative of other problems with the Component Broker interaction, and there is no way to be certain whether or not a deadlock condition has occurred when your application encounters one of these symptoms.

Each of the following values used in your Component Broker application program must be set consistently:

### **DB2 database lock timeout**

Sets the time limit for how long a DB2 agent (application) waits on a lock before the DB2 lock manager terminates the DB2 request. When this occurs an SQL0913N error (-913) is returned to the DB2 application (the Component Broker application server). Component Broker then propagates this condition back to the Component Broker application by raising either a `CORBA::PERSIST_STORE` or `CORBA::TRANSACTION_ROLLEDBACK` exception, depending upon the processing being done within Component Broker for that application's request. The default DB2 database lock timeout is infinite (-1). The DB2 lock timeout may be set using the `UPDATE DATABASE CONFIGURATION` statement within DB2 or by setting it using the DB2 Control Center. The lock timeout should be set large enough to provide sufficient processing time for any application that holds the lock being waited for, but small enough that an application caught in a deadlock condition will return within its specified guidelines.

### **Component Broker client style's ORB request timeout**

Sets the time limit for how long a Component Broker application waits for its ORB request to be processed and responded to by the appropriate Component Broker server. When the ORB request timeout is reached, a `CORBA::NO_RESPONSE` exception is raised and thrown to the Component Broker application. The default Component Broker ORB request timeout is 30 seconds but can be set by using the Component Broker Systems Management EUI, and editing the appropriate client style image or application server image used by the Component Broker application. The ORB request timeout should be large enough to allow sufficient time for the DB2 lock timeout to occur and for Component Broker to process the condition internally and appropriately respond to the application.

### **Component Broker transaction timeout**

Sets the time limit for how long a Component Broker transaction remains idle or waits on an action to complete. When the Component Broker transaction timeout is reached, Component Broker's transaction service rolls back the offending transaction and notifies the Component Broker application by raising a `CORBA::TRANSACTION_ROLLEDBACK` exception. The default Component Broker transaction timeout is 5 minutes (300 seconds), and is set using the `set_timeout()` method on the application's transaction current object. The Component Broker transaction timeout should be large enough to allow sufficient time for the DB2 lock timeout to occur and for Component Broker to process the condition internally.

The recovery code in your Component Broker application should include rolling back the current transaction (in the case of catching either `CORBA::NO_RESPONSE` or `CORBA::PERSIST_STORE`), doing any application specific clean up, and then attempting to retry the transaction. Since any of the three exceptions mentioned could also indicate other Component Broker problems, you should limit the number of retry attempts.


---

## **Heap Size**

Under some conditions the application Heap Size for an individual database needs to be increased.

If you are seeing SQL error code -954 in the CBCConnector activity log or in the db2diag log you need to increase the DB2 Application Heap Size.

Details on how and when to do this can be found in "Chapter 14, Operational Performance" in the *DB2 Administrative Guide*. See the "How DB2 Uses Memory" topic for information about the `DB2 applheapsz` parameter.

 To configure the DB2 database from the Control Center, perform the following steps:

1. On the Windows NT **Start** menu, select **Programs** → **DB2 for Windows NT** → **Administration Tools** → **Control Center**.
2. Locate the Database that you want to reconfigure. Select **Open up system name** → **Instances** → **Node name** → **Databases**.
3. On the pop-up menu for the database:
  - Open the Wizard:
    - a. Click **Configure performance** to open the Performance Configuration Wizard.
    - b. Follow the instructions in the Wizard to determine the correct heap size.
  - Open the Configure Database GUI:
    - a. Click **Configure** to open the Configure Database GUI.
    - b. Click the **Performance** tab.
    - c. Change the heap size.

---

## DB2 Transaction Logs

The DB2 database saves information about active transactions in a set of log files. In the default configuration of DB2, where `LOGRETAIN = OFF` and `USEREXIT= OFF`, these log files are reused in a circular fashion as transactions are successfully committed or rolled back. However, when a transaction is not successfully ended by either being committed or rolled back (for example, a client system is powered-off in the middle of a transaction), DB2 does not reuse the log files. This has the effect of the log files containing more and more data as other transactions, from other clients, continue to be processed until the unfinished transaction is either committed or rolled back. If the unfinished transaction does not complete before the log is filled up, DB2 will stop processing requests from other clients. These requests will be rejected with `SQLCODE -964` (transaction log full) errors.

Component Broker is built to tolerate unreliable clients and has provided a mechanism to allow other clients to continue processing while a single client may have a transaction that has not been completed. This is done with a timeout value being supplied to the transaction service. However, there is some interaction between the value set in the timeout, the amount of data being modified in each transaction, and the typical length of time that a transaction takes to complete. If these have been balanced appropriately, the unfinished transaction will be rolled back and the activity log will contain corresponding messages. If, however, these have not been balanced appropriately, the activity log will contain messages indicating that the DB2 transaction log has filled up, and `PERSIST_STORE` exceptions will be returned on client requests.

The following configuration values are provided for reference:

```
LOGFILSIZ = 2500 (2500 4K pages = 10 MB per log file)
LOGPRIMARY = 3 (number of primary log files)
LOGSECOND = 32 (number of secondary log files)
```

These settings are sufficient to run applications with transaction timeouts of five minutes, where the server throughput is near two hundred transactions per minute, and each transaction is executing about twenty database operations. Applications that vary from this can alter these configuration settings to tune the system to match their usage requirements. The trade-offs are between the amount of disk space needed for the logs, how long the transactions run, and how many database operations each transaction executes.

Elimination of transaction log full errors is possible with the following actions:

1. Decreasing the transaction timeout value proportionally decreases the amount of log space required. Conversely, increasing the transaction timeout value proportionally increases the amount of log space required.

2. Decreasing the aggregate work throughput of the Component Broker server proportionally decreases the amount of log space required. Conversely, increasing the aggregate work throughput of the server proportionally increases the amount of log space required.
3. Decreasing the number of DB2 database operations (that is, inserts, updates, and so on) done by each transaction proportionally decreases the amount of log space required. Conversely, increasing the number of DB2 database operations done by each transaction proportionally increases the amount of log space required.



---

## Appendix C. Installing a Sample Configuration Using UDB

In this section discusses procedures to create a five-machine configuration. The configuration consists of the following:

- Host1 - UDB Server
- Host2 - Component Broker System Manager
- Host3 - Component Broker Server with UDB Client (same user ID as Host1)
- Host4 - Component Broker Server with UDB Client (different user ID than Host1)
- Host5 - Component Broker Client

For more information, see the *IBM DB2 Universal Database for Windows NT Administration Getting Started Version 5*.

In the following procedures:

- **Node** refers to any unique node you want to create.
- **Hostuid** refers to the ID you want to use for a particular Host.
- **Password** refers to the password for that user ID.

---

### Install and Configure Host1

Set this computer up as a DB2 server:

1. Insert the DB2 v5.0 for Windows NT compact disc into your CD-ROM drive. If your system autoloads the compact disc, you can skip the next two steps.
2. Display the contents of the compact disc.
3. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To start the installation, you must click the Setup icon that looks like a computer.
4. In the Welcome window, click the **Next** button.
5. Select DB2 Universal Database Enterprise Edition.
6. Click the Typical Install installation choice and select the installation drive and directory.
7. Type the user name: *Host1uid* with ID: *password* (if you're on a domain, type a domain-qualified user ID with password).
8. After the installation completes, restart your computer.
9. When the First Steps panel is displayed, create the SAMPLE database.
10. Open the Services folder and verify that the DB2 - DB2DAS00 service is running. If not, click on Startup, type your password, then confirm your password by retyping it.
11. Set DB2 Security Server to automatic.
12. Add the DB2COMM system environment variable with a value of TCPIP.
13. Add the following lines to the `c:\WINNT\system32\drivers\etc\services` file:

```
DB2 3700/tcp
DB2I 3701/tcp
```
14. From DB2 CLP, enter:

```
update database manager configuration using svcname DB2
```
15. Restart your computer.

16. If you are developing Component Broker applications, you need to install the SDK.
17. Insert the DB2 v5.0 Software Developer's Kit compact disc. If your system autoloads the compact disc, you can skip the next two steps.
18. Display the contents of the compact disc.
19. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To start the installation, you must click the Setup icon that looks like a computer.
20. When installing the SDK, ensure that the **Administering Remote Servers** check box is selected.
21. When the installation completes, restart your computer.

---

## Install Host2

Perform a custom installation to install the following packages on this computer:

- System Manager
- System Manager User Interface

Follow the procedure documented in "Installing the Custom Install Option" on page 103. When asked to supply the user ID and password for the Component Broker Service, type the Host2uid with its password.

**Note:** When installing a System Manager, you do not need to have DCE or UDB on the machine.

---

## Install and Configure Host3

Install the UDB Client on the computer you have identified as Host3:

1. Insert the DB2 v5.0 Client Application Enablers compact disc. If your system autoloads the compact disc, you can skip the next two steps.
2. Display the contents of the compact disc.
3. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To start the installation, you must click the Setup icon that looks like a computer.
4. Click the **Next** button in the Welcome Window.
5. Select the **Install components required to administer remote servers** check box.
6. Click the **Typical Install** option and select the destination drive and directory.
7. When the installation completes, restart your computer.
8. In the Services folder, set DB2 Security Server to automatic.
9. Add the DB2COMM system environment variable with a value of TCPIP.
10. Add the following lines to the `c:\WINNT\system32\drivers\etc\services` file:  

```
DB2      3700/tcp
DB2I     3701/tcp
```
11. From DB2 CLP, enter:  

```
catalog tcpip node node remote Host1.rchland.ibm.com server DB2
```

DB2 corresponds to the the svcname given when configuring the UDB server.
12. Catalog the Sample database created on Host1. Enter:  

```
catalog database sample as sample at node node
```
13. Connect to the Sample remote database. Enter:



connect to sample user *Host3uid* using *password*

Where *Host3uid* is the same as the one used on Host1. This verifies that you can access databases on the server.

14. Restart your computer.
15. If you are developing Component Broker applications, you need to install the SDK.
16. To install SDK insert the DB2 v5.0 Software Developer's Kit compact disc. If your system autoloads the compact disc, you can skip the next two steps.
17. Display the contents of the compact disc.
18. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To start the installation, you must click the Setup icon that looks like a computer.
19. When the installation completes, restart your computer.

Install the Component Broker Server. Perform a custom installation to install the following packages on this computer:

- Server
- System Manager Agent

Follow the procedure documented in "Installing the Custom Install Option" on page 103. Supply the following information during the installation and configuration:

- In the Enter Remote DB2 Server Information window, enter the following information:

**User ID** *Host3uid*.

**Password** The password associated with the user ID.

**Server Host** The fully-qualified name of the remote server (Host1). For example:

*Host1.rchland.ibm.com*

**Local Node** The node for this computer.

- In the Input User ID and Password For Component Broker Service window, enter the following information:

**User ID** *Host3uid*.

**Password** The password associated with the user ID.

- In the Enter System Management Host Name and Port Number window, enter the following information:

**Host** The fully-qualified name of the host on which the System Manager is installed, in this case, Host2. Enter: *Host2.rchland.ibm.com*

**Port** The default port number is 20002. Accept the default.

**Note:** When installing a Server, you need both DCE and UDB.

In the DB2 CLP at the end of the install, you will see some errors because the script is for UDB servers, and this is a client. Perform the following steps:

1. Open up a new db2 CLP

2. Enter:

*attach to node user Host3uid using password*

This will attach to the node you want the database created on.

3. Enter:

```
create database CBSRFCDB
```

This creates the reference collections database on the specified node

4. Enter:

```
connect to CBSRFCDB user Host3uid using password
```

Now all tables created will be qualified with *Host3uid*.

5. Enter:

```
create table RCTABLE (KEY INTEGER NOT NULL,  
HANDLE VARCHAR(1000) FOR BIT DATA NOT NULL)
```

6. Enter:

```
create table RCHMTABLE (KEY VARCHAR(250) FOR BIT DATA NOT NULL PRIMARY KEY,  
TYPEID VARCHAR(500) FOR BIT DATA NOT NULL)
```

7. Enter:

```
create table KRCTABLE (KEY INTEGER NOT NULL, HANDLE VARCHAR(1000)  
FOR BIT DATA NOT NULL, KEYDATA VARCHAR(1000) FOR BIT DATA NOT NULL )
```

8. Enter:

```
bind x:\CBroker\etc\DBCConnector.bnd
```

9. Enter:

```
bind x:\CBroker\etc\KeyedDBCConnector.bnd
```

10. Enter:

```
bind x:\CBroker\etc\ICollectionsBOIMDOImpl_I.bnd
```

11. Enter:

```
connect reset
```

**Note:** You can create a script to do the previous steps for you, modeled after `inscrt.db2` in the `x:\CBroker\etc` directory.

12. Reboot your computer.

---

## Install and Configure Host4

**Note:** Since *Host4uid* is different than *Host1uid*, you must create a Windows NT user on Host1 that corresponds to the *Host4uid*. It must have full administrator rights.

Install the UDB Client:

1. Insert the DB2 v5.0 Client Application Enablers compact disc. If your system autoloads the compact disc, you can skip the next two steps.
2. Display the contents of the compact disc.
3. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To start the installation, you must click the Setup icon that looks like a computer.
4. Click the **Next** button in the Welcome Window.
5. Select the **Install components required to administer remote servers** check box.
6. Click the **Typical Install** option and select the destination drive and directory.
7. When the installation completes, restart your computer.
8. In the Services folder, set DB2 Security Server to automatic.

9. Add the DB2COMM system environment variable with a value of TCP/IP.
10. Add the following lines to the c:\WINNT\system32\drivers\etc\services file:
 

```
DB2      3700/tcp
DB2I     3701/tcp
```
11. From DB2 CLP, enter:
 

```
catalog tcpip node node remote Host1.rchland.ibm.com server DB2
```

 DB2 corresponds to the the svcname given when configuring the UDB server.
12. Catalog the Sample database created on Host1. Enter:
 

```
catalog database sample as sample at node node
```
13. Connect to the Sample remote database. Enter:
 

```
connect to sample user Host4uid using password
```

 Where *Host4uid* is the user ID for Host4 which is different from the one on Host1 and Host3
14. Restart your computer.
15. If you are developing Component Broker applications, you need to install the SDK.
16. To install the SDK, insert the DB2 v5.0 Software Developer's Kit compact disc. If your system autoloads the compact disc, you can skip the next two steps.
17. Display the contents of the compact disc.
18. Start the installation by clicking the Setup icon. Three Setup icons are displayed in the contents list. To start the installation, you must click the Setup icon that looks like a computer.
19. When the installation completes, restart your computer.

Install the Component Broker Server. Perform a custom installation to install the following packages on this computer:

- Server
- System Manager Agent

Follow the procedure documented in "Installing the Custom Install Option" on page 103. Supply the following information during the installation and configuration:

- In the Enter Remote DB2 Server Information window, enter the following information:

**User ID** *Host4uid*

**Password** The password associated with the user ID.

**Server Host** The fully-qualified name of the remote server (Host1). For example:

Host1.rchland.ibm.com

**Local Node** The node for this computer.

- In the Input User ID and Password For Component Broker Service window, enter the following information:

**User ID** *Host4uid*.

**Password** The password associated with the user ID.

- In the Enter System Management Host Name and Port Number window, enter the following information:

**Host** The fully-qualified name of the host on which the System Manager is installed, in this case, Host2. Enter: Host2.rchland.ibm.com

**Port** The default port number is 20002. Accept the default.

**Note:** When installing a Server, you need both DCE and UDB.

In the DB2 CLP at the end of the install, you'll see some errors. Perform the following procedure:

1. Open up a new db2 CLP.

2. Enter:

```
connect to CBSRFCDB user Host4uid using password
```

Now all tables created will be qualified with *Host4uid*.

3. Enter:

```
create table RCTABLE (KEY INTEGER NOT NULL,  
HANDLE VARCHAR(1000) FOR BIT DATA NOT NULL)
```

4. Enter:

```
create table RCHMTABLE (KEY VARCHAR(250) FOR BIT DATA NOT NULL PRIMARY KEY,  
TYPEID VARCHAR(500) FOR BIT DATA NOT NULL)
```

5. Enter:

```
create table KRCTABLE (KEY INTEGER NOT NULL,  
HANDLE VARCHAR(1000) FOR BIT DATA NOT NULL,  
KEYDATA VARCHAR(1000) FOR BIT DATA NOT NULL )
```

6. Enter:

```
bind x:\CBroker\etc\DBConnector.bnd
```

7. Enter:

```
bind x:\CBroker\etc\KeyedDBConnector.bnd
```

8. Enter:

```
bind x:\CBroker\etc\ICollectionsBOIMDOImpl_I.bnd
```

9. Enter:

```
connect reset
```

**Note:** You can create a script to do the previous eight steps for you, modeled after inscrt.db2 in the `x:\CBroke\etc` directory.

10. Reboot the machine

---

## Install Host5

Install the Component Broker client. Perform a custom installation to install the following packages on this computer:

- CORBA Client
- System Manager Agent

Follow the procedure documented in "Installing the Custom Install Option" on page 103. Supply the following information during the installation and configuration:

- In the Enter Client's Bootstrap Host Name and Port Number window, enter the following information:

**Host** Type the fully-qualified bootstrap host for the client. For example, you could use either of the following:

– Host3.rchland.ibm.com

– Host4.rchland.ibm.com

**Port** The default port number is 900. Accept the default.

- In the Input User ID and Password For Component Broker Service window, enter the following information:

**User ID** *Host5uid.*

**Password** The password associated with the user ID.

- In the Enter System Management Host Name and Port Number window, enter the following information:

**Host** The fully-qualified name of the host on which the System Manager is installed, in this case, Host2. Enter: Host2.rchland.ibm.com

**Port** The default port number is 20002. Accept the default.

**Note:** When installing a Client, you do not need to have DCE or UDB on the computer.

---

## Summary

The complete configuration has now been installed.

Following are a few more hints for developing and running applications on this configuration:

1. Since the client can have only one bootstrap host, and this example uses multiple Component Broker servers, any factory finders being done in the code need to be workgroup scoped instead of host scoped. For example, the following line:

```
obj = CBSeriesGlobal::nameService()->
    resolve_with_string("host/resources/factory-finders/host-scope");
```

would be changed to:

```
obj = CBSeriesGlobal::nameService()->
    resolve_with_string("workgroup/resources/factory-finders/workgroup-scope");
```

2. When running a client application, you must make some changes in the EUI in order for the databases to be accessed:
  - a. Expand Management Zones → *myhost* → Configurations → *myconfiguration* → RDB Connections and select *mydatabasemanager1*.
  - b. From the pop-up menu for *mydatabasemanager1*, select **Edit**.
  - c. Select the **Main** Tab
  - d. Edit the **openString** attribute box and add the user ID and password for connecting to the database. For example, if the **openString** attribute box reads ClaimDB, change it to ClaimDB, user ID, and password.
  - e. Repeat step 2d for each database in this folder.
  - f. Reactivate your configuration to populate these changes into your host and server images.

Here are some general tips for the UDB Server/UDB Client relationship:

1. To access a database from a UDB Client machine, one of the following conditions must be true:
  - The database needs to be created from that client.
  - The database needs to be cataloged from that client.

2. If you want to access a database from a UDB client using a different user ID than the UDB server, you must create that user ID as a member of the Windows NT Users group on the UDB server. You can then connect to the database using the new ID. Any subsequent table you create under the new connection will be your new user ID's table.
3. When performing a Component Broker server install, the CBSRFCDB database is created when the install runs inscrt.db2. When a database is created, you need to create binds for it. For CBSRFCDB, the files that need to be bound to that database are x:\CBroker\etc\DBConnector.bnd, x:\CBroker\etc\CollectionsBOIMDOImpl\_I.bnd, and x:\CBroker\etc\KeyedDBConnector.bnd.
4. For any user created databases, x:\CBroker\etc\db2cntcs.bnd and x:\CBroker\etc\db2cntrr.bnd need to be bound if using the cache service.
5. If you want to bind a database (as is done in the previous example), you must create that user ID as a member of the Windows NT administrators group on the UDB server.

---

## Appendix D. The Two-Compiler Environment (NT only)

Installing the ActiveX client SDK and the C++ client SDK on the same system can present conflicts involving headers, includes, and executable files. These conflicts exist in both the run-time environment code and the development environment code. To run in a two-compiler environment, you need to take the following precautions:

- Create a separate directory for the ActiveX header files.
- Define a separate environment for each compiler.

**Note:** If you install the ActiveX client run time or the ActiveX client SDK, you need to create a two-compiler environment to avoid compile errors with the VisualAge C++ compiler.

---

### Creating Header Directory for ActiveX Client

You need to create a separate directory to contain several Component Broker headers supplied with the ActiveX client SDK. By default, the installation stores the following ActiveX client SDK files in the `x:\CBroker\include` directory.

- `ctype.h`
- `limits.h`
- `stdio.h`
- `stdlib.h`
- `string.h`
- `time.h`
- `sys/types.h`
- `wchar.h`
- `windows.h`

These header files use the `MS_INC_DIR` variable in their implementation and are intended only for use with the Microsoft Visual C++ compiler.

**Note:** If these header files are used with the VisualAge C++ compiler, errors occur.

To create a separate, searchable directory:

1. Create the `VCPP` subdirectory under `x:\CBroker\include`.
2. Move the specified ActiveX header files to this directory.
3. Modify the `INCLUDE` variable as discussed in "The Microsoft Visual C++ Environment" on page 190

---

### Preparing Separate Environments

Working in a two-compiler environment requires a separate environment for each compiler. You can create several batch (`.bat`) files to accomplish this. When you need a specific compiler-environment, you can run the appropriate batch file in a command window to create the specific environment. In this manner, you can create a window for each compiler-environment.

## The VisualAge C++ Environment

For the VisualAge C++ environment, the following suggestion for the INCLUDE and PATH variables may help:

- Modify the INCLUDE variable by removing any reference to the Microsoft Visual C++ directories.
- Ensure that the INCLUDE variable contains the `x:\CBroker\include;` statement before the VisualAge C++ directories statement.
- Modify the PATH variable by removing any reference to the Microsoft Visual C++ directories.

## The Microsoft Visual C++ Environment

For the Microsoft Visual C++ environment, the following suggestions for the INCLUDE and PATH variable may help:

- Modify the INCLUDE variable by removing any reference to the VisualAge C++ directories.
- Modify the INCLUDE variable by adding the new directory you created for the ActiveX headers. For example:

```
set INCLUDE=x:\CBroker\include;  
          x:\CBroker\include\VCPP;  
          x:\DevStudio\VC\include;%INCLUDE%
```

- Ensure that the `x:\CBroker\include` and the `x:\CBroker\include\VCPP` directories are before any reference to the Microsoft Visual C++ directories.
- Modify the PATH variable by removing any reference to the VisualAge C++ directories.



---

## Appendix E. Enabling the Security Service

Security enablement settings can only be changed before the server process or the client process is brought up. After a server or a client is brought up, the security enablement setting in the System Management Common Data Store cannot be changed until the server or the client is brought down.

**Note:** SSL security is available only on Windows NT.

Use the following procedures to enable the Security Service for Name Servers and DCE C++ Clients (CORBA or ActiveX) through the System Manager User Interface. If the Security Service is already enabled see “Disabling Security Service” on page 151 to disable it before proceeding.

### Important

- You need to stop your servers (or clients) before changing your security setting to turn security either on or off. See Appendix J, “Stopping Servers” on page 215 for details. Updates to an attribute in a server image (instead of the preferred method of changing and activating the configuration) may deactivate a running server.
- If any Component Broker server is configured with security enabled, the other Component Broker servers in the same Component Broker configuration should also be configured with security enabled. The procedure to enable security for a Component Broker server is described in the following sections.
- You are encouraged to read the “Administer Security in your Enterprise” topic in the *Component Broker for Windows NT and AIX System Administration Guide* to learn more about Security in Component Broker.

---

## Enable DCE Security for Servers in a Server Group

This procedure assumes that your application is added to a server that is contained in a server group. In the System Manager User Interface, server groups are located in the Management Zone.

1. From the System Manager User Interface, select **View** → **User Level** → **Expert**.
2. Enable the Security Service:
  - a. Select the server group:
    - For the Name Servers:**  
Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → Name Server, and select Sample Name Server.
    - For the Application Servers:**  
Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → Server Groups, and select your server group.
  - b. Open the Object Editor:
    - For the Name Servers:**  
Open the pop-up menu of Sample Name Server and select **Edit**. This opens the Object Editor.
    - For the Application Servers:**  
Open the pop-menu of your server group and select **Edit**. This opens the Object Editor.
  - c. Click the **Security Service** tab.

- d. Change the **security enabled** field from no to yes.
- e. Click the **OK** button.

---

## Enable DCE Security for the C++ Client in a Management Zone

1. Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → Client Styles, and select *myhost.austin.ibm.com* Default Client.
2. Open the pop-up menu of *myhost.austin.ibm.com* Default Client, and select **Edit**. This opens the Object Editor.
3. Select the **Security Service** tab.
4. Change the **security enabled** field from no to yes.
5. Select the **Orb** tab.
6. Change the **Request timeout** field from 30 to 300.
7. Click the **OK** button to exit the window.

---

## Activate DCE Security Enablement for Servers and Clients

1. From the System Manager User Interface, select **View** → **User Level** → **Expert**.
2. Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration.
3. Open the pop-up menu for the Sample Configuration and select **Activate**.

**Note:** Settings you make using this procedure are temporary and are lost the next time you activate the application.

---

## Enable SSL Security for Servers in a Server Group

This procedure assumes that your application is added to a server that is contained in a server group. In the System Manager User Interface, server groups are located in the Management Zone. In addition, DCE security must be activated for the servers in your server group as detailed in “Activate DCE Security Enablement for Servers and Clients.” If you have not yet activated DCE security for the servers in your server group, do so before continuing.

1. From the System Manager User Interface, select **View** → **User Level** → **Expert**.
2. Enable the Security Service.

a.

### For the Name Server:

Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → Name Server, and select Sample Name Server.

### For the Application Server:

Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration → Server Groups, and select your server group.

b.

### For the Name Server:

Open the pop-up menu for Sample Name Server, and select **Edit**. This opens the Object Editor.

**For the Application Server:**

Open the pop-menu for your server group, and select **Edit**. This opens the Object Editor.

- c. Select the **Security Service** tab.
- d.  Change the **SSL Type-I Client Association Enabled** field from no to yes.
- e.  Ensure that **Standard Export QOP Model** is set up properly.

**Note:** When choosing a Quality of Protection (QOP) value, “Confidentiality” provides the highest level of message protection, “Integrity” provides somewhat less protection, and “Authenticity” provides the lowest level of message protection. See the “Security Administration” topic in the *Component Broker for Windows NT and AIX System Administration Guide* for detailed definitions of the Standard Export-QOP values.

- f. Select the **OK** button.

**Note:** You can temporarily override the server group security setting for an individual server. Expand Host images → *myhost* → Server Images *myserver* and use the same procedure that you would use to enable security for a server group (in this section). This customization is lost the next time you activate the application.

---

## Activate SSL Security Enablement for Servers in a Server Group

1. From the System Manager User Interface, select **View** → **User Level** → **Expert**.
2. Configure the server keyring file for each server image. Expand Host Images → *myhost* → Server Images.
3. Open the pop-up menu for Server Images and select **Edit**. This opens the Object Editor.
4. For each configured server image:

- Select the Security tab and edit the SSL Keyring file field. (Note a default path and keyring file name have already been defined for each configured server image. You can select to use the defaults or create new path and name. To change, edit the SSL Keyring file field.)

You can also select to use the default server keyring file “CBDevTest.kdb” or create your own server keyring file (Please note that “CBDevTest.kdb” keyring file is provided for test only. Using this file with any application that is in deployment or production, will lead to serious security exposure. Refer to the “Use the Test Certificate Provided by Component Broker” section of the *Component Broker for Windows NT and AIX System Administration Guide* for more details.)

- If you select to use the default server keyring file (CBDevTest.kdb), do the following:
  - a. Do not edit the SSL Keyring file field.
  - b. Go to the directory defined in the SSL Keyring file field.
  - c. Inside the directory, you should find CBDevTest.kdb.
  - d. Copy CBDevTest.kdb to the default file name that is defined in the SSL Keyring file field.
- If you select to use your own server keyring file, do the following:
  - a. Define the new path and file name by editing SSL Keyring field.
  - b. Make sure the newly-defined path and file name does exist.

5. Expand Management Zones → Sample Cell and Work Group Zone → Configurations → Sample Configuration.
6. Open the pop-up menu of the Sample Configuration and select **Activate**.

---

## Enable Java Client With DCE Security

By default the security service is disabled for Java clients.

A client can indicate to the Java ORB whether to enable DCE security in an application or an applet. The parameters can be specified in the code, where `CBSeriesGlobal.initialize` is called. The following sample illustrates how the ORB parameters can be passed using the `CBSeriesGlobal.initialize` call:

```
...
java.util.Properties props = new java.util.Properties();
props.put("org.omg.CORBA.ORBClass", "COM.ibm.CORBA.iiop.ORB");
props.put("com.ibm.CORBA.BootstrapHost", "server.rchland.ibm.com");
props.put("com.ibm.CORBA.BootstrapPort", "900"); // optional, 900 is default
props.put("com.ibm.CORBA.EnableDCESecurity", "true");
//NOTE different signature for applets vs applications
CBSeriesGlobal.Initialize(this, props); // For an applet (pass the applet object)
CBSeriesGlobal.Initialize(args, props); // For an application (pass the args array)
...
```

There is an additional way in which security can be enabled. This procedure differs depending on whether it is for an application or for an applet.

The application programmer can set a system property by specifying the `-D` option to the Java interpreter. For example, to run `PolicyApp` with Security enabled, say:

```
java -Dcom.ibm.CORBA.EnableDCESecurity=true
      -Dcom.ibm.CORBA.BootstrapHost=server.rchland.ibm.com
      -Dcom.ibm.CORBA.BootstrapPort=900 PolicyApp
```

System properties cannot be accessed in this case, so the applet programmer must pass properties on `CBSeriesGlobal.Initialize()` (as stated previously) or must specify applet parameters in the HTML file under the `APPLET` tag to have them passed into the ORB as attachments to the applet object (which is passed as the first argument to `CBSeriesGlobal.Initialize()`). The following is an example of passing ORB parameters using an HTML file:

```
...
<applet code = Samples.StackApp.class
  width = 400
  height = 150>
  <param name=org.omg.CORBA.ORBClass
    value="COM.ibm.CORBA.iiop.ORB">
  <param name=com.ibm.CORBA.BootstrapHost
    value="server.rchland.ibm.com">
  <param name=com.ibm.CORBA.BootstrapPort
    value="900">
  <param name=com.ibm.CORBA.EnableDCESecurity
    value="true">
</applet>
...
```

The property is first looked for in the System properties (default values).

Then the property parameters are used (the `props` argument in `CBSeriesGlobal.Initialize`); they overwrite any parameters set in the system properties.

Then the command line arguments (or applet arguments) are used; they overwrite any arguments set in the property parameters.

---

## Enable Java Client With SSL Security

By default, the Security Service is disabled for Java client. SSL Security can be enabled by setting the `ClientStyleImageURL` parameter to a client style properties file that has security enabled. For initial testing purposes only, Component Broker provides a default properties file (`CBDevTestCIDefault.properties`), a server key ring file (`CBDevTest.kdb`), and a Java Client key ring class (`CBDevTestCIKeyRing.class`). The server key ring file and Java Client key ring class both contain a self-signed site certificate. These key ring files are provided as a convenience only and should not be used when you deploy your application. You should take the time to formulate a policy for establishing a trust-basis for your enterprise and acquire and distribute the server certificates accordingly. See the “Security Administration” section of the *Component Broker for Windows NT and AIX System Administration Guide* for more details about establishing a trust-basis and administering certificates.

**Note:** SSL is based on cryptographic key algorithms for authentication and data protection. An SSL Java client has a key ring class to specify its public key and the SSL server has a key ring file to contain its private key. For information about generating the public and private keys see the “Security Administration” section of the *Component Broker for Windows NT and AIX System Administration Guide*.

To use the default properties file substitute the `CBDevTestCIDefault.properties` for the `ClientStyle1.properties` file in the following examples. The `CBDevTestCIDefault.properties` file enables SSL Type-I Security and uses the `CBDevTestCIKeyRing.class` as the key ring class for the Java Client.

## Create Your Own Client Properties File

If you use the default client properties file, skip this section.

You can create a new properties file for your client by following these steps:

1. Expand Management Zones → Sample Cells and Work Group Zone → Configurations → Sample Configuration → Client Styles, and select your client or `ClientStyle1`.
2. Open the pop-up menu of your client, and select **Edit**. This opens the Object Editor.
3. Select the **Security Service** tab.
4. Change the **Security enabled** field from `no` to `yes`.
5. Change the **SSL TYPE-I Server Association Enabled** from `no` to `yes`.
6. Set up the proper value for **SSL keyring file**, **SSL keyring password**, **Standard Perform QOP Model**, **SSL V3 session timeout**, and **SSL credential timeout** fields. The key ring file specified must be a Java class created from a key ring which contains the certificates of the servers with which the Java Client expects to communicate. If the key ring class is protected by a password, it must also be declared in the **SSL keyring password** field. See the “Security Administration” topic of the *Component Broker for Windows NT and AIX System Administration Guide* for more information about these fields.
7. Select the **Orb** tab.
8. Change the **Request timeout** field from `30` to `300`.
9. Click the **OK** button to exit the window.
10. The properties files with a name of *Client-Image-Name.properties* will be generated in the *CB-BASE-DIR/data/properties* directory (where *CB-BASE-DIR* is the Component Broker CDS data directory specified in the NT Regedit database).

You must place this file on the web server for access by a Java applet or on the client machine for access by a Java application.

## Enable Application or Applet with SSL Security

For both applications (mains) and applets, system properties (such as ClientStyleImageURL) can be passed to the Object Request Broker (ORB) from the Java program. The following is an example of how these properties can be passed:

```
java.util.Properties props=new.java.util.Properties();
props.put("org.omg.CORBA.ORBClass", "com.ibm.CORBA.iiop.ORB");
props.put("com.ibm.CORBA.BootstrapHost", "server.rchland.ibm.com");
props.put("com.ibm.CORBA.BootstrapPort", "900");
// optional, 900 is the default
props.put("com.ibm.CORBA.ClientStyleImageURL",
         "file:c:\CBroker\data\properties\ClientStyle1.properties");
// Set to the path and name of the client style properties file
// you want to use.
// For an applet(pass the applet object)
//NOTE different signature for applets vs applications
    CBSeriesGlobal.Initialize(this, props); // For an applet (pass the applet object)
    CBSeriesGlobal.Initialize(args, props); // For an application (pass the args array)
...

```

For an application, any system property can be passed by specifying it with the `-D` option to the Java interpreter. For example, to run PolicyApp with security enabled, enter:

```
java -nojit -Dcom.ibm.CORBA.ClientStyleImageURL=
      "file:c:\CBroker\data\properties\ClientStyle1.properties"
      -Dcom.ibm.CORBA.BootstrapHost=server.rchland.ibm.com
      -Dcom.ibm.CORBA.BootstrapPort=900 PolicyApp

```

### Notes:

1. The above examples only work in the case where the Java run time can read local files.
2. There are many other forms of URLs, all of which can be used here.
3. The `-nojit` option turns off the Java "just-in-time" compiler option.

For an applet, the properties can be passed using the `APPLET` tag.

```
<applet code = Samples.StackApp.class width=400 height=150>
<param name=org.omg.CORBA.ORBClass value="com.ibm.CORBA.iiop.ORB">
<param name=com.ibm.CORBA.BootstrapHost value="server.rchland.ibm.com">
<param name=com.ibm.CORBA.BootstrapPort value="900">
<param name=com.ibm.CORBA.ClientStyleImageURL=
      "http://web-server-host/properties/ClientStyle1.properties">
</applet>

```

**Precedence rules::** The property is first looked for in the System properties (default values).

Then the property parameters are used (the props argument on CBSeriesGlobal.Initialize); they overwrite any parameters set in the system properties.

Then the command line arguments (or applet arguments) are used; they overwrite any arguments set in the property parameters.

For additional information on security enablement with Java clients, see the Component Broker online documentation.

---

## Troubleshooting Security Problems

When running applications with security enabled, the following checklist provides useful information in isolating and troubleshooting security problems.

### 1. Check the DCE configuration and run-time status.

Because Component Broker supports only one DCE cell, ensure that all DCE clients and their DCE servers are configured into one cell.

WIN

Perform the following procedure:

- a. From the Windows NT **Start** menu, select **Programs** → **DCE for Windows NT** → **DCEsetup**.
- b. In the right panel, ensure that the "Cell Name:" entry is consistent with the same entry on the other hosts within the cell.

- c. For a configured DCE server, ensure that the services of:

- Security Client
- Security Server
- CDS Advertiser
- CDS Server
- DTS

are **enabled** and in **running** state.

- d. For a configured DCE client, ensure that the services of:

- Security Client
- CDS Advertiser
- DTS

are **enabled** and in **running** state.

AIX

Perform the following procedure:

- a. From a shell prompt, enter:

```
lsdce
```

to list all configured DCE services. Each service should be in COMPLETE state.

- b. Enter:

```
dce_login
```

to ensure that the DCE services are running.

- c. Enter:

```
getcellname
```

to return the DCE cell name for each host. Ensure that all hosts have the same cell name.

### 2. Check the security enablement status of the client or server.

From the System Manager User Interface, check all entries under Client Style Images and Server Images. For each entry listed under Client Style Images and Server Images, open its pop-up menu and select **Edit**. In the Object Editor, select the **Security Service** tab, and ensure that the **security enabled** field is set to yes.





---

## Appendix F. File Locations and the Java Classpath

Component Broker includes the following key environments that run Java code. Each of these has specific requirements relating to its use of .jar and .zip files.

**Java Client** The Java client and the Java client SDK leverage a Java ORB that is represented by somojor.zip. The server SDK leverages some of the same parsing tools as the Java server SDK. The Java client does not leverage IOM.

**Component Broker Server** The Component Broker run time does not use the same ORB technology as the Java client. Therefore the presence of the Java client run time support is not only unnecessary, but in fact creates problems when it is included in the classpath. The IOM facility within Component Broker provides implementations (in fact is a wrapper in Java to the C++ ORB and a great deal more) of the same ORB interfaces that are supported on the Java client. There is a necessary common set of interfaces supported on the Java client and on the Component Broker server. This makes the programming model work. Java client code, for example, can easily be cut and pasted into a Java BO implementation and will work fine. Under the covers, this sort of code is handled much differently on the server than on the Java client.

The somojor.zip and somshor.zip files are mutually exclusive. The Java client must use somojor.zip and cannot use somshor.zip; Java in the server must use somshor.zip and cannot use somojor.zip. The main reason that the client and server sides each uses a different classpath is that the IOM stubs are incompatible with Java ORB stubs. Since the names of the stubs are OMG architected, the stubs have to be separately emitted and packaged, one for the client side and one for the server side.

**Note:** Work is under way to standardize the stubs to allow the same stubs to be used on both the server and client sides. This should eliminate the classpath problem.

Before you run a Java client application on a computer on which you have the Component Broker server run time installed, you need to add the following files to the beginning of your classpath (only in the window in which you run your client application, not in the system or user environment classpath):

- somojor.zip, which contains classes to support the client-side Java ORB. If this is not at the beginning of the classpath, the wrong classes will be found, and your application will not run.
- The JAR files that contain your Java client bindings (located in the JCB subdirectory, with the naming convention JCBMyObjectC.jar). These contain classes to support a client application accessing the equivalent Java component on the server.

In addition to these runtime environments, the SDK environments include a large number of .jar files. The Object-Level Trace and Remote Debug features of Component Broker are written in Java and work on both the Java client and on the Component Broker Server. Some of the required .jar files are in the classpath by default. Others get pulled into the classpath using server specific startup files.

Object Builder is on the server SDK classpath as it is normally configured. The classpath does not normally include all of the Object Builder .jar files because they have their own startup file (ob.bat) which sets the environment before launching the tool. Only .jar files common to Object-Level Trace, Remote Debug, and ObjectBuilder are put into the classpath at install time.

The somojor.zip file is placed on the server SDK image so that Object Builder-generated Java client bindings can be created from a server SDK image. To test Java client applications from the server SDK, you should create a .bat file that places somojor.zip into the classpath for the duration of the application or for the life of the command shell. this prevents the server from seeing the somojor.zip file and lets it work correctly.

The following table lists the .zip and .jar files shipped with Component Broker, where they are needed, and for what they are used.

**Notes:**

1. In the table:
  - **CB** represents **Component Broker**.
  - **CP** represents **Classpath**.
  - **OLT** represents **Object Level Trace**.
  - **OB** represents **Object Builder**.
  - **MOFW** represents **Managed Object Framework**.
2. The Component Broker run time is a prerequisite for the Component Broker server SDK.
3. The Component Broker Java client run time is a prerequisite for the Component Broker Java client SDK.
4. Additional .jar and .zip files can be found on the system when using PAA or when using the samples that are shipped with Component Broker.

Zip File	CB Server Run Time	CP	CB Server SDK	CP	CB Java Client Run Time	CP	CB Java Client SDK	CP
ibmcbjs.zip (IOM Java Server Stubs for Object Services and MOFW)	Yes	Yes	Gets from server run time	Yes	No	No	No	No
somib01.zip (Interface Repository)	Yes	No	Gets from server run time	No	No	No	No	No
somshcl.zip (IDL C++ emitters + IOM Java emitters)	Yes	Yes	Gets from server run time	Yes	No	No	No	No
somshor.zip (IOM Java run time - including IOM Java stubs for C++ ORB interface)	Yes	Yes	Gets from server run time	Yes	No	No	No	No
somojj.zip (idl parser tools)	No	No	Yes	Yes	No	No	Yes	Yes
somojr.zip (Java client run time)	No	No	Yes - here for building	No	Yes	Yes	Gets from Java client run time	Yes
ivbtrol.jar (OLT Java viewer)	No	No	Yes	No	No	No	Yes	No
ivbtrutil.jar (OLT Java common code)	No		Yes	No	No	No	Yes	No
ivbtrjrt.jar (run time for OLT)	No	No	Yes	Yes	No	No	Yes	Yes

Zip File	CB Server Run Time	CP	CB Server SDK	CP	CB Java Client Run Time	CP	CB Java Client SDK	CP
ivbtrc.jar (OLT Java client controller)	No	No	Yes	No	No	No	Yes	No
ivbjdebug.jar (OLT Java Debugger)	No	No	Yes	Yes	No	No	Yes	Yes
ivbjfaceall.jar (JFace - GUI Framework)	No	No	Yes	No	No	No	No	No
ivbjfc.jar ("Swing" run time)	No	No	Yes	No	No	No	No	No
ivbob.jar (OB)	No	No	Yes	No	No		No	No
ivbsop.jar (OB)	No	No	Yes	No	No		No	No
ivxml.jar (IBM XML parser)	No	No	Yes	No	No		No	No
somjse.zip (SSlight Java library for SSL)	No	No	No	No	Yes	Yes	Gets from Java client run time	Yes
ivbbrg.jar (OB)	No	No	Yes	No	No	No	No	No
ivbclone.jar (OB)	No	No	Yes	No	No	No	No	No

## Additional Java Considerations

Component Broker clients and servers need specific versions of the JDK. These versions could vary depending on whether or not Java is being used for business objects, procedural adaptor support, or client applications. The previous section of this appendix outlines which levels are required on which configurations. This section includes some general pointers about Java levels:

1. The Component Broker installation programs generally check that the installed level of the JDK matches the documented prerequisite level. This means, for example that the installation program verifies that JDK 1.1.6 or later is available for a Component Broker server installation. Later versions are usually allowed, but are not supported. Check the Late Breaking News for support statements about newer versions of the JDK.
2. The Component Broker installation programs attempt to ensure that the right version of the JDK is in the PATH. However, it should be noted that as other software packages are installed on the same system as Component Broker, additional versions of Java might be installed with them. In some cases, the software packages' installation programs add these versions of the JDK to the front of the PATH.

Putting the recommended JDK level for Component Broker in the front of the system environment variable for PATH is the safest way to proceed. Exactly how the other software packages deal with user and system environment variables does vary, however, and each case must be examined individually. Also remember that if you are trying to edit the PATH variables together manually, you may be forced to link the user PATH with the system PATH by placing a %PATH% at the end of your user PATH variable.



---

## Appendix G. Java and Just in Time (JIT) in Component Broker

This appendix includes the following topics:

- “Using JIT on Component Broker for Windows NT”
- “Using JIT on Component Broker for AIX” on page 204
- “Using JIT on Component Broker for OS/2 and Windows 95” on page 205

You can use the following methods to disable the JIT on AIX, Windows NT, Windows 95, and OS/2:

- Set the JAVA\_COMPILER environment variable to an empty string (works only for the java command).
- Use the -nojit option on the java or jre command.
- Pass -Djava.compiler= on the command line for either the java or the jre commands.

When controlled using the JVM settings in Component Broker Systems Management, including java.compiler= in the properties field disables the JIT. Deleting java.compiler= from the properties field enables the JIT.

For Component Broker, you should not explicitly disable JIT. You should allow each Component Broker Component to determine whether the use of JIT is allowed or not. This is illustrated in the tables in the following sections..

The tables in the following sections describe the usage of Java and JIT by particular components in Component Broker. The **Default** column indicates how this component will run and which settings are used if you do not alter the Component Broker defaults. The **Options** column indicates whether or not the default can be altered and still have the component work properly. The **Comments** column explains other things related to JIT and Java as it relates to the particular Component Broker Component.

---

### Using JIT on Component Broker for Windows NT

**Note:** JDK 1.1.6 is a prerequisite for Component Broker for Windows NT.

In general, you should allow usage of the JIT on NT. The default install of JDK 1.1.6 is enabled to run JIT. This means that all Java applications will use JIT, unless it is explicitly disabled.

**Note:** Versions of the JDK prior to 1.1.6 required you to install the performance pack to get JIT enabled.

Component Broker Component	Default	Options	Comments
Java Application client	Uses JAVA_COMPILER settings.	Runs either way.	
Java Application client with SSL	You must launch the application with -nojit.	None.	JIT compiler problem.
Java BO (assumes you are not using CICON backed PO/PAO)	java.compiler= is part of JVM settings in Systems Management.	Modify System Management settings to run with JIT on.	Little performance gain by turning it on, but it works fine.
Enterprise Java Beans	java.compiler= is part of JVM settings in Systems Management.	Modify System Management settings to run with JIT on.	Little performance gain by turning it on, but it works fine.

Component Broker Component	Default	Options	Comments
Any PAA backed business object	java.compiler= is part of JVM settings in Systems Management.	None.	Little performance gain by turning it on, but it works fine.
Object Builder	Uses JAVA_COMPILER settings	Put -nojit in ob.bat if you want to override	Runs much better with it turned on. Some minor problems have been seen (for example, screen repainting) that get better with the Windows NT JIT 1.1.6 fixpack installed. (See note 1)
Quick Test	The Quick Test samples save the original JAVA_COMPILER setting, set it to NONE, and after emitting or running, reset it to the original setting	Put -nojit when you start Quick Test to suppress usage	Quick Test emitters do not work with JIT on
idlc emitters	JIT is enabled	Use -nojit to suppress usage	No known problems

**Notes:**

1. This can be acquired from the Component Broker Support Lab Web pages.

Next, consider how Web browsers on the Windows NT platform deal with the JIT.

Browser	Java Applet Client Application	Java Applet Client Application with SSL Enabled	Quick Test Run Time in an Applet
Netscape 4.07	Works fine. There is no option available to turn off the JIT.	Works fine. There is no option available to turn off the JIT.	Only if the JDK 1.1.6 patch or upgrade is applied, or a plug-in is used.
Microsoft Internet Explorer 4.01 SP1	Runs either way.	Runs either way. The performance gain is two times greater with the JIT.	Only if the plug-in is used.

**Note:** Running Java in the Web browser using the Java plug-in has not been tested extensively. However, Quick Test runs on both Netscape 4.07 and Internet Explorer 4.01 with the Java plug-in.

## Using JIT on Component Broker for AIX

**Note:** JDK 1.1.6, level a116-19980924 is used by Component Broker for AIX.

You should not use the JIT on AIX.

Component Broker Component	Default	Options	Comments
Java Application client	Uses JAVA_COMPILER settings.	Runs either way.	
Java Application client with SSL	Not supported by Component Broker 2.0.	n/a	n/a
Java BO	java.compiler= is part of JVM settings in System Management.	Modify System Management settings to run with JIT on.	Little performance gain by turning it on, but it works fine.
Any PAA backed business object	java.compiler is part of JVM settings in System Management.	Modify System Management settings to run with JIT on.	Little performance gain by turning it on, but it works fine. IMS/HOD is not supported on AIX for Component Broker 2.0.
Object Builder	Uses JAVA_COMPILER settings.	Put -nojit in ob.sh if you want to override.	There is little difference in performance on AIX with or without the JIT. There are also problems importing XML with the JIT enabled.
Quick Test	Uses environment settings for Java which should not have the JIT set on.	Put -nojit when you start Quick Test to suppress usage.	Some strange behaviors with focus events on text entry fields, but if an OOSQL predicate drop-down is selected, it behaves fine after that.
idlc emitters	JIT is disabled.	None. JIT cannot be enabled on AIX.	USING JIT causes random failures and slows down idlc by 100%.

Next, consider how Web browsers on the AIX platform deal with the JIT. There are no JIT options on the Netscape browser.

Browser	Java Applet Client Application	Java Applet Client Application with SSL Enabled	Quick Test Run Time in an Applet
Netscape Navigator 4.07 (JDK 1.1.5)	Supported.	Not supported in Component Broker 2.0.	Results are not known at this time (See note 1).
<b>Notes:</b>			
1. Consult the <i>Late Breaking News</i> and the support lab for up to date information regarding this topic.			

## Using JIT on Component Broker for OS/2 and Windows 95

**Note:** These are run time-only clients that do not have any SDK options.

Component Broker for OS/2 uses JDK 1.1.6. You can obtain the JDK 1.1.6 at:

[http://service.boulder.ibm.com/asd-bin/doc/en\\_us/java116/f-feat.htm](http://service.boulder.ibm.com/asd-bin/doc/en_us/java116/f-feat.htm)

Component Broker for Windows 95 uses JDK 1.1.6.

Component Broker Component	Default	Options	Comments
Java Application client - Windows 95	Uses JAVA_COMPILER settings	Runs either way	
Java Application client - OS/2	Uses JAVA_COMPILER settings	See note 1	See note 1
Java Application client with SSL - Windows 95	See note 1	See note 1	See note 1
Java Application client with SSL - OS/2	See note 1	See note 1	See note 1
<b>Notes:</b>			
1. Consult the <i>Late Breaking News</i> and the support lab for up-to-date information regarding these topics.			

Next, consider how Web browsers on the Windows 95 and OS/2 platforms deal with the JIT.

Browser	Java Applet Client Application	Java Applet Client Application with SSL Enabled	Quick Test Run Time in an Applet
Netscape 4.07 (Windows 95)	Works fine. There is no available option to turn off JIT.	See note 1.	Not supported in Component Broker 2.0.
Internet Explorer 4.01 SP1 (Windows 95)	Runs either way.	See note 1.	Not supported in Component Broker 2.0.
Netscape 4.04 (OS/2)	Works fine. There is no available option to turn off JIT.	See note 1.	Not supported in Component Broker 2.0.
<b>Notes:</b>			
1. Consult the <i>Late Breaking News</i> and the support lab for up to date information regarding these topics.			

**Microsoft Internet Explorer 4.01 Configuration Notes:** You may see messages in the Java console regarding Component Broker file access errors and security exceptions. These are not fatal and the applet should work successfully. However this logging does slow down execution of the applet. To alleviate this logging, do the following:

1. Start Internet Explorer.
2. On the menu bar, select **View** → **Internet Options**.
3. Click the **Security** tab.
4. In the **Zone** field, select the zone you want to change from the drop-down list (Local intranet zone, Internet zone, and so on).
5. In the **Internet zone** box, select the **Custom** radio button.
6. Click the **Settings** button.
7. On the Security Settings window, scroll to Java and select Java permissions → custom.
8. Click the **Java Custom Settings** button.
9. On the Internet zone window, click the **Edit Permissions** tab.
10. Under Unsigned content, select the following:

**Access to all files:** Enable



**Access to all network addresses:** Enable

**System information:** Enable

11. On the Internet zone window, click the **OK** button.
12. On the Security Settings window, click the **OK** button.
13. On the Internet Options window, click the **Apply** button, then the **OK** button.
14. Restart Internet Explorer.



---

## Appendix H. Environment Variables

For Component Broker to run correctly, platform specific environment variables must be properly set during installation.

---

### Windows NT Platform

The Component Broker installation creates a set of environment variables in the System Environment settings. For the user environment space, the following rules apply:

- If the environment variable does not exist in the user space:
  - If the environment variable is one that can be shared by other products, then the environment variable is created in the user space with a reference to its system environment variable. For example:  
`NLSPATH=%NLSPATH%`
  - If the environment variable is CB specific such as SOMCBASE, then no additional settings are required in the user space. The user environment variable is picked up for the system environment setting value.
- If the environment variable exists in the user space, then check if there is a reference to its system environment variable:
  - If yes, no additional setting is required except for the LIB and INCLUDE user environment variables. For the LIB user environment variable, the path to %SOMCBASE%\LIB is prepended. For the INCLUDE user environment variable, the path to %SOMCBASE%\INCLUDE is prepended.
  - If no, a reference to its system environment variable is prepended. During uninstall of the Component Broker product, the reference to the system environment variable is not unset.

---

### AIX Platform

The setting of Component Broker environment variables are controlled by the sourcing of the CBConnector.profile file.

---

### Environment Variables

The following subset of environment variables are those with which you may need to be familiar to get Component Broker up and running. Each has a brief description. Some may not apply to your environment.

Parameter	Operating System	Description
SOMCBASE	NT, AIX	Directory where the Component Broker run time is installed.
SOMCBENV	NT, AIX	Used by the Component Broker run time to determine which configuration image should be used.
CL	NT	Sets a Visual C++ compiler option that is needed by the Component Broker development environment.
CLASSPATH	NT, AIX	On Windows NT, CLASSPATH is constructed using CB_*_CP environment variables.
DPATH	NT	Used by the VisualAge C++ compiler to search for message catalogs.
EXTHSM	AIX	Extends the number of shared memory segments needed to run Component Broker.
HELP	NT	Sets the directory where the Component Broker Help files are located.
ICC	NT	Sets VisualAge C++ compiler options that are needed by Component Broker.
IDLC_INCLUDE	NT, AIX	Sets the directory where the Component Broker header files used by Object Builder are located.
IDLC_OPTIONS	NT	Used by the Component Broker idlc executable.
INCLUDE	NT	Used by the compiler and debugger. It specifies the path the compiler and debugger use to locate C++ include files (files included in your source code with the #include directive). The %SOMCBASE%\INCLUDE directory must come before the compiler include directory.
IVB_DGB_PORT	NT	This variable specifies the default port number on which the Object Level Trace(OLT) daemon is listening for a connection. This defaults to 8001 if not given and is overridden by values specified in the OLT configuration.
IVB_DRIVER_PATH	NT, AIX	Directory where the Component Broker Toolkit and samples are installed.
LIB	NT	Lists directories that the linker searches for library. The %SOMCBASE%\LIB directory should be listed before the compiler lib directory.
LIBPATH	AIX	Lists the directories to search for Component Broker shared libraries.
LD_LIBRARY_PATH	AIX	Lists the directories to search for Component Broker Java shared libraries.
LOCPATH	NT, AIX	Used to locate Locale data.
LOCPATHCBM	NT	Used to locate ActiveX specific locale data.
NLSPATH	NT, AIX	Used for National Language Support.
PATH	NT	Lists the directories to be searched for executable files.
SMTMP	NT	Sets the directory where the Component Broker System Management places all temporary work files.
VABHELP	NT	Used by the Component Broker Online Documentation.

The following environment variables do not need to be set to get Component Broker up and running. They can be set to help overcome any problems with APPC communications to tier-3 systems, usually on advice from your IBM support personnel. These environment variables override default APPC behaviour.

SOMCB_APPC_LIBRARY_NAME	NT, AIX	Specifies the name of the IBM Communication Server APPC DLL which is used to send or receive requests over SNA. Its default value is wappc32.dll. If the full pathname of the DLL is not specified, the first library found with this name from directories in the PATH is loaded.
SOMCB_NOF_LIBRARY_NAME	NT, AIX	Specifies the name of the IBM Communication Server Node Operations Facility DLL which is used to monitor the status of the SNA node and create the SNA configuration. Its default value is winnof32.dll. If the full pathname of the DLL is not specified, the first library found with this name from directories in the PATH is loaded.
SOMCB_APPC_PARTNER_LIVES	NT, AIX	Specifies the number of times a server should be restarted before it deletes an APPC Connection that is neither defined in the server's configuration nor required to complete outstanding transactions. The default value is infinity.
SOMCB_APPC_TIMEOUT	NT, AIX	Specifies the maximum number of seconds an allocate or receive request is permitted to take. If these operations take longer than the timeout value then they are cancelled. The default value is 120 seconds (2 minutes). A value of 0 means infinite (or no time limit).
SOMCB_APPC_DEFAULT_MODE_NAME	NT, AIX	Specifies that the SNASVCMG mode group should be used for exchange log names (XLN) requests at server startup.



---

## Appendix I. Starting and Restarting Servers

Servers are normally started when the configuration that defines them is activated. This is controlled by the run control attribute of each Server Group Model and Server Model in the Configuration. After their Configuration is activated, individual servers can be stopped and restarted as needed. Normally, the servers would be restarted by reactivating the Configuration using the Activate command. The Activate command ensures that all the required servers are started properly. Manually restarting servers is only suggested as an occasional method for restarting individual servers.

To restart a server, activate its Configuration:

1. From the System Manager User Interface menu, select **View** → **User Level** → **Expert**.
2. From the System Manager User Interface Home view, expand Host Images → *myhost.austin.ibm.com* → Server Images, and select the host name you want to start.
3. Open the pop-up menu, and select one of the following:

**Run** Starts the server when it receives an ORB method request.

**Run Immediate**

Starts the server immediately. It is ready to process its first ORB method request without delay.

**Note:** If a server fails to start, a message is displayed in the Event Log indicating that the server is not able to run. To enable the server to run, follow this procedure again.





---

## Appendix J. Stopping Servers

**Note:** Before stopping the name server, ensure that any application servers are stopped.

If you need to stop a server:

1. From the System Manager User Interface menu, select **View** → **User Level** → **Expert**.
2. From the System Manager User Interface Home view, expand Host Images → *myhost.austin.ibm.com* → Server Images, and select the server name you want to stop.
3. Open the pop-up menu, and select one of the following:

### **Stop**

Stops the server with full shut down processing. All user-tasks on the server are completed, and the required information is written to the system management logs.

### **Stop Quick**

Stops the server with partial shut down processing. Use this action only if the Stop action failed. This action does not wait for user-tasks to be completed.

### **Stop Immediate**

Stops the server without shut down processing. Use this action only as a last resort.

**Note:** Once a server is stopped, it cannot be restarted until it is placed in a run state. To place a server in a run state, see Appendix I, “Starting and Restarting Servers” on page 213.

If you need to shut down a Component Broker server in an untimely or forced manner, you may need to manually clean up DB2 connections and locks that were obtained. If these DB2 connections and locks are left on your DB2 database, they can interfere with future application usage.

To manually clean up after an untimely or forced shut down, the following tasks may need to be done to prevent future interaction problems between the server and DB2. Perform these tasks from a DB2 command prompt, if necessary.

From a DB2 command prompt:

1. Enter:

```
force applications all
```

All connections from the server to DB2 are broken, and all locks held by flight transactions are freed.

2. Enter:

```
list indoubt transactions with prompting
```

If any locks are held by in doubt transactions, they are listed numerically. For each lock held, commit or rollback the transaction. For example, to commit the first transaction in the list, enter:

```
c 1
```

To rollback the second transaction, enter:

```
r 2
```

For additional information on these commands, see the *DB2 Command Reference* installed with DB2. For in-depth information on manual recovery, see the *DB2 Administration Guide*.



---

## Appendix K. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Department LZKS  
11400 Burnet Road  
Austin, TX 78758  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

---

## Trademarks

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

CICS  
AIX  
DB2  
IBM  
MVS/ESA  
OS/2  
OS/390  
PowerPC  
VisualAge

AFS and DFS are trademarks of Transarc Corporation in the United States, or other countries, or both.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Oracle and Oracle8 are registered trademarks of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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





Part Number: C092798

Printed in U.S.A.

