Component Broker for Windows NT and AIX

# System Administration Guide

*Release 2.0*

SC09-2704-03

Component Broker for Windows NT and AIX

# System Administration Guide

*Release 2.0*

SC09-2704-03

**Fourth Edition (January 1999)**

This edition applies to release 2.0 of IBM Component Broker for Windows NT and AIX, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Consult the latest edition of the applicable IBM system bibliography for current information on this product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

# Contents

# What's New

This edition is based on the *System Administration Guide* for release 1.3 of IBM Component Broker, SC09-2704-02. The following main changes have been made from that previous edition:

**General Restructure**
This book has been generally restructured to try and make it easier for you to create new Component Broker enterprises. This restructure also separates the main areas of system administration, to enable you to concentrate on those areas that you are interested in.

**"Chapter 3. Create a New Single-Host Enterprise" on page 85**
New information about how to create a new Component Broker single-host environment (of hosts, cell, work groups, and so on). This host environment is typically for testing Component Broker and applications and as a base for application development.

**"Chapter 4. Create a New Multi-Host Enterprise" on page 101**
New information about how to create a new Component Broker multi-host environment (of hosts, cell, work groups, and so on). This host environment is for application deployment.

**"Chapter 5. Configure a new Application Environment" on page 133**
New information about how to configure a new application environment (of application servers, client styles, applications, and so on). This describes configuring application environments for use in single- and multi-host environments.

**"Chapter 6. Administer your Host Environment" on page 167**
Information about managing and changing your host environment. This builds on information provided in the earlier chapters; "Chapter 3. Create a New Single-Host Enterprise" on page 85 and "Chapter 4. Create a New Multi-Host Enterprise" on page 101.

**"Chapter 7. Administer Application Servers" on page 181**
Information about managing and changing your server groups and freestanding servers, including how to configure new services and characteristics. This builds on information provided in the earlier chapter "Chapter 5. Configure a new Application Environment" on page 133.

**"Chapter 8. Administer Clients" on page 205**
Information about managing and changing your client styles, including how to configure new services and characteristics. This builds on information provided in the earlier chapter "Chapter 5. Configure a new Application Environment" on page 133.

**"Chapter 9. Administer Applications" on page 221**
Information about managing and changing applications to run on your server groups, freestanding servers, and client styles, including how to configure new services and characteristics. This builds on information provided in the earlier chapter "Chapter 5. Configure a new Application Environment" on page 133.

**"Chapter 10. Administer Management Zones and Configurations" on page 253**
Information about using Management Zones and their Configurations to extend or change the organization of your host environment or application environment. This builds on information provided in the earlier chapters; "Chapter 3. Create a New Single-Host Enterprise" on page 85, "Chapter 4.

Create a New Multi-Host Enterprise" on page 101, and "Chapter 5. Configure a new Application Environment" on page 133.

**"Chapter 11. Administer Security in your Enterprise" on page 261**
Information about making your enterprise secure with Component Broker, and about Component Broker's use of DCE and SSL security.

**"Chapter 12. Administer Workload Management" on page 333**
Information about implementing workload management of controlled server groups. This builds on information provided in the earlier chapter "Chapter 5. Configure a new Application Environment" on page 133.

**"Chapter 13. Administer Connections to Tier-3 Systems" on page 343**
Information about how to configure APPC, ECI, and HOD connections to tier-3 systems, such as CICS and IMS, and databases such as DB2 and Oracle. This builds on information provided in the earlier chapter "Chapter 5. Configure a new Application Environment" on page 133.

**"Configure Communications Server" on page 367**
This now includes new information about how to configure IBM Communications Server for AIX to enable APPC communication between Component Broker application servers on AIX and remote systems across a SNA network.

**"Chapter 14. Administer Component Broker Services" on page 397**
New information about configuring and managing Component Broker services used by application servers, client styles, and applications. This builds on information provided in the earlier chapter "Chapter 5. Configure a new Application Environment" on page 133.

**"Chapter 15. Operate your Enterprise" on page 413**
Information about operating the runtime state of your enterprise, including starting and stopping application servers and monitoring runtime information generated by system management.

**"Appendix A. Features of the User Interface" on page 433**
General information about the windows and functions of the System Manager user interface. Detailed information is provided by the context-sensitive help information of the user interface.

**"Configure a Chain of Location Objects" on page 234**
A new task topic about how to configure a chain (ordered list) of location objects to be used by a factory finder to look for a factory that supports a desired type of managed object.

**"Appendix B. Quality of Protection" on page 453**
Reference information about the quality of protection (QOP) attributes used to configure security for client styles, servers, and host daemons.

**"Appendix C. Related Topics" on page 469**
Several topics related to system administration that do not fall within the normal administration areas described in the body of the system management information.

**Bibliography**
A new bibliography topic with lists of books and sources of information related to Component Broker system management, but provided separate from Component Broker.

These topics contain links to related concept and task information. Links are also provided from the Table of Contents.

# About This Book

The *IBM Component Broker System Administration Guide* contains information about System Management, including the System Manager, System Management Agent, System Manager User Interface, and the DDL Editor. It describes the model, concepts, and tasks to help you define, configure, and operate things to be managed by Component Broker System Management.

## Who Should Read This Book

This book is for administrators and others who are responsible for administering system enteprises running under Component Broker.

## How this Book is Organized

"Chapter 1. An Overview of Administering Component Broker Enterprises" on page 1 provides an overview of what you should do to adminster Component Broker enterprises. It introduces the main stages of system administration and describes the functions and components of system management, the parts of your enterprise, and how the System Manager represents your enterprise.

"Chapter 2. Use the System Manager User Interface" on page 55 describes how to start the System Manager user interface and to use it to find and act on objects displayed. This tells you how to use the System Manager user interface generally, and forms a set of general information that you use when completing real administration tasks through the System Manager user interface. For example, the description of editing objects is often used when configuring objects to be managed, as described later in this book.

"Chapter 3. Create a New Single-Host Enterprise" on page 85 describes the concepts and procedures used to create a standalone host for testing Component Broker and as a base for an application development host. For a more realistic enterprise in which to deploy your application environment, see "Chapter 4. Create a New Multi-Host Enterprise" on page 101.

"Chapter 4. Create a New Multi-Host Enterprise" on page 101 describes the concepts and procedures used to create a multi-host enterprise for deploying an application environment across the managed hosts.

"Chapter 5. Configure a new Application Environment" on page 133 describes the concepts and procedures used to configure an application environment for use on a single- or multi-host environment (configured in the preceding sections).

"Chapter 6. Administer your Host Environment" on page 167 provides information about administering hosts in an existing Component Broker host environment (configured in the preceding sections). It describes the concepts and procedures used to configure and operate hosts, cells, work groups, and their communication protocols and name servers.

"Chapter 7. Administer Application Servers" on page 181 provides information about administering application servers in an existing application environment (created in "Chapter 5. Configure a new Application Environment" on page 133). It describes the concepts and procedures used to configure and operate application servers.

"Chapter 8. Administer Clients" on page 205 provides information about administering clients in an existing application environment (created in "Chapter 5. Configure a new Application Environment" on page 133). It describes the concepts and procedures used to configure and operate client styles.

"Chapter 9. Administer Applications" on page 221 provides information about administering applications in an existing application environment (created in "Chapter 5. Configure a new Application Environment" on page 133). It describes the concepts and procedures used to configure and operate applications for use on application servers and client styles.

"Chapter 10. Administer Management Zones and Configurations" on page 253 provides information about administering management zones and their configurations in existing host and application environments (created in earlier sections).

"Chapter 11. Administer Security in your Enterprise" on page 261 provides information about administering security in existing host and application environments (created in earlier sections). It describes the concepts and procedures used to configure and operate Component Broker security.

"Chapter 12. Administer Workload Management" on page 333 provides information about administering workload management in an existing application environment (created in "Chapter 5. Configure a new Application Environment" on page 133). It describes the concepts and procedures used to configure and operate Component Broker workload management.

"Chapter 13. Administer Connections to Tier-3 Systems" on page 343 provides information about administering connections to tier-3 systems (for example, CICS and IMS) in an existing application environment (created in "Chapter 5. Configure a new Application Environment" on page 133). It describes the concepts and procedures used to configure and operate HOD, ECI, APPC, and RDB connections to tier-3 systems.

"Chapter 14. Administer Component Broker Services" on page 397 provides information about administering Component Broker services to add or change base services in an existing application environment (created in "Chapter 5. Configure a new Application Environment" on page 133). It describes the concepts and procedures used to configure Component Broker services.

"Chapter 15. Operate your Enterprise" on page 413 describes the concepts and procedures used to monitor and act on the runtime configuration of your enterprise.

"Appendix A. Features of the User Interface" on page 433 provides information about the System Manager user interface. It describes the windows and functions of the user interface. More detail is provided as contextual help for the System Manager user interface.

"Appendix B. Quality of Protection" on page 453 provides information about the qualities of protection (QOP) That the Component Broker security service uses to protect messages that flow between clients and servers.

Related Topics provides information about some concepts and tasks that are related to system management. This information is provided to help you with system management.

″Index″ provides an index of system management subjects, as an alternative way to access topics in this book.

## 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, user output, and directory paths. |
| **bold** | Menu choices, push buttons, check boxes, radio buttons, group-box controls, drop-down list boxes, combo-boxes, notebook tabs, and entry fields. |
| *italics* | Programming keywords, variables, and attributes, titles of information units, initial use of unique terms, and emphasis. |

## Notation

The term *CBConnector* is an abbreviation for *Component Broker Connector*, the runtime part of Component Broker.

New or unusual terms are generally described when first used, but if you have difficulty with some terms, please refer to the *Component Broker Glossary*, SC09-2710-00. You can view the Glossary online by clicking **Glossary** on the home page of the Component Broker web.

## Component Broker Information

The following information is part of Component Broker:

- Help information is available from Component Broker product panels.
- The Component Broker online library can be viewed using a frames-compatible Web browser.
- *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*, 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*, SC09-2707 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.

# Chapter 1. An Overview of Administering Component Broker Enterprises

This topic provides an overview of what you should do to administer a Component Broker enterprise. It aims to introduce the main stages that you should complete. Related Concepts provide more information about the functions and components that you can use to administer a Component Broker enterprise.

Your enterprise can be considered as two parts, the *host environment* and the *application environment*, as shown in Figure 1 on page 2.

**Host environment**
This comprises the System Manager host, all server and client-only hosts that it manages, the hosts' name servers and protocols, the Component Broker cell and work groups, and any remote name contexts linking to other system name trees.

**Application environment**
This comprises your applications, the application servers and client styles that applications run on, and application resources like DLLs, containers, and connections to tier-3 systems.

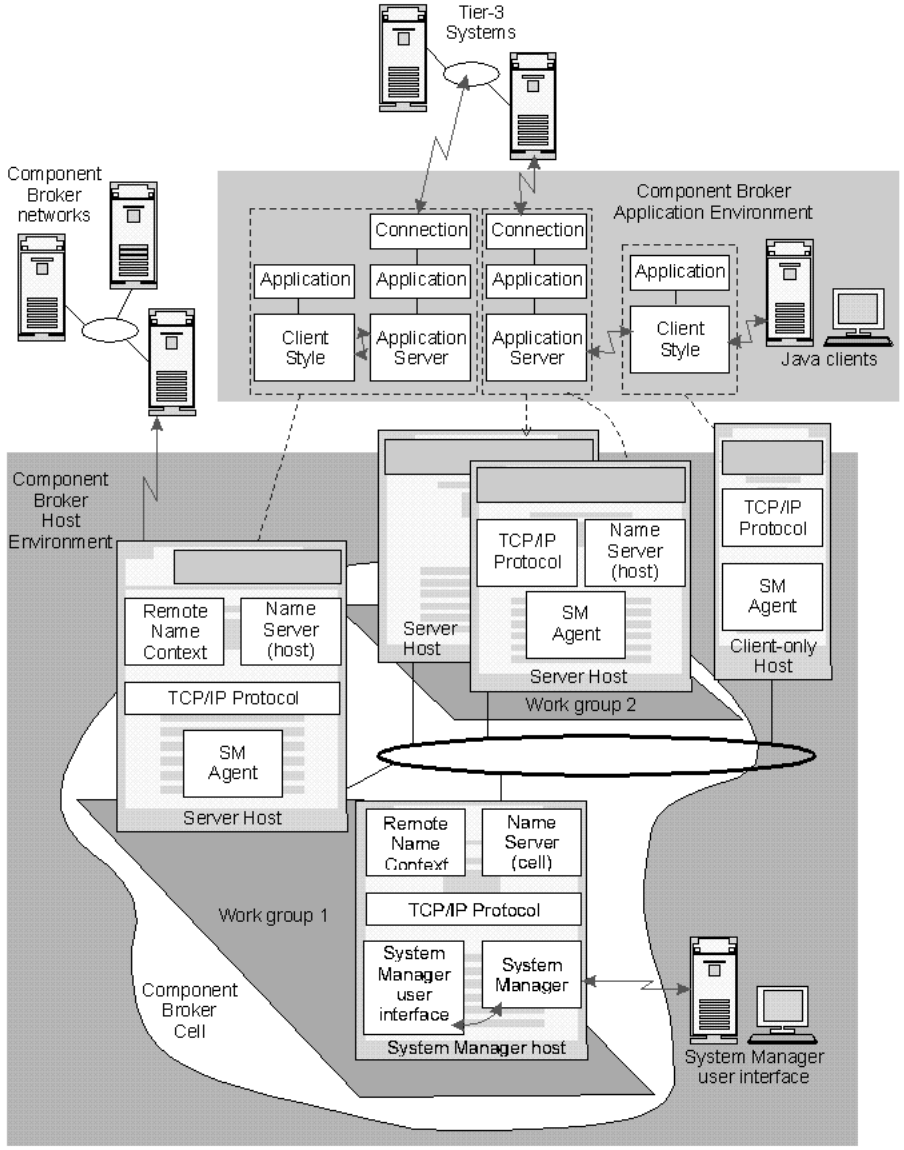


*Figure 1. Component Broker Host and Application Environments*

For more information about the host and application environments, see “Parts of your Enterprise Managed by Component Broker” on page 15.

### What Should you do to Administer a Component Broker Enterprise?

To administer a Component Broker enterprise, you complete the following general tasks:

1. **Install appropriate Component Broker software and prerequisites on the hosts that form your host environment.** This is described in the *Planning, Performance, and Installation Guide*.

   For an overview of installing Component Broker, see the following topics:

   - “Install a Component Broker Single-Host Environment” on page 88.
   - “Install a Component Broker Multi-Host Environment” on page 105

2. **Configure the host environment.** This defines and activates the initial configuration of the hosts managed by Component Broker, its system name tree, cell, and work groups. You configure your host environment in one *management zone*, which should be reserved for only the host environment.
   - You can create a single-host Component Broker enterprise, for testing and as a base for application development. The standalone workstation used has all the Component Broker system management functions and your application environment.

     For information about how to do this, see “Chapter 3. Create a New Single-Host Enterprise” on page 85.
   - To deploy applications, you create a multi-host Component Broker enterprise as illustrated in the above figure. The System Manager runs on a separate host, which also provides the name service for the Component Broker cell and the minimum work group required by Component Broker. Other managed *server hosts* are used to support applications running on application servers, and usually support client styles that access those applications. Managed *client-only hosts* can be used to support separate client styles that access applications running on other server hosts.

     For more information about how to do this, see “Chapter 4. Create a New Multi-Host Enterprise” on page 101

   In both single- and multi-host environments, you should define your host environment in one *management zone*. Within each management zone you can define one or more alternative *Configurations*.
3. **Configure the application environment.** This defines and activates the initial configuration of the applications, application servers, and client styles to run on Component Broker hosts. As part of this task, the System Manager automatically distributes your applications (and their resources) to the managed server hosts and client-only hosts on which the applications are to run.

   You configure your application environment in one or more management zones, separate from the management zone reserved for the host environment. Separating the host environment and application environment in this way optimizes the configuration of your enterprise and the ease with which you can administer it. Within each management zone you define one or more alternative *Configurations*.

   For more information about how to do this, see “Chapter 5. Configure a new Application Environment” on page 133.
4. **Operate your enterprise**; for example, to activate complete Configurations, subscribe to events within your enterprise, and stop or start individual application servers.

   For more information about how to do this, see “Chapter 15. Operate your Enterprise” on page 413.
5. **You can also extend or change your enterprise**; for example, to add new hosts or workgroups to your host environment, to add applications to your application environment or configure them onto new server groups, or to add extra functions like workload management or tier-3 communications.

   For more information about how to do this, see the following topics:
   - “Chapter 6. Administer your Host Environment” on page 167
   - “Chapter 7. Administer Application Servers” on page 181
   - “Chapter 8. Administer Clients” on page 205

- “Chapter 9. Administer Applications” on page 221
- “Chapter 10. Administer Management Zones and Configurations” on page 253
- “Chapter 11. Administer Security in your Enterprise” on page 261
- “Chapter 12. Administer Workload Management” on page 333
- “Chapter 13. Administer Connections to Tier-3 Systems” on page 343

The remainder of this topic provides a general description of the system management functions, components, and object model used to manage Component Broker enterprises. For more information, see the following topics:

- “What CBConnector System Management Provides”
- “Components used for System Management” on page 9
- “Parts of your Enterprise Managed by Component Broker” on page 15
- “System Management Representation of Your Enterprise” on page 37

## What CBConnector System Management Provides

This topic describes what CBConnector System Management provides to help you manage your enterprise:

**“Installation of Software” on page 5**
Local installation and removal of CBConnector software and centralized installation and removal of CBConnector application software.

**“Administration of Enterprises” on page 6**
Definition of and action on host computers, servers, clients, and applications. This includes defining logical groupings of hosts, servers, clients and applications to organize and simplify the management of your enterprise.

**“Operating your Enterprise” on page 6**
Action on the servers, clients, and applications that exist on host computers within your enterprise, including starting and stopping servers and applications.

**“Centralized Configuration of System Management Objects” on page 7**
To administer your enterprise, you use the System Manager user interface to configure a network of system management objects. The System Manager stores the configuration data centrally and manages copies of that data where needed on managed hosts around your enterprise. Therefore, you can use the System Manager user interface as a single-point-of-control to centrally create all the system management objects for your enterprise, and to centrally operate your enterprise.

**“Monitoring the Health of Application Servers” on page 7**
System Management monitors automatically the health of application servers that it manages and can display the health of a server to give you an indication of how loaded the server is and whether or not it is short of resources.

**“Integrated System Management with Tivoli” on page 8**

The Tivoli system management product provides a number of features for managing large enterprises. Component Broker system management can be integrated with the Tivoli system management product through a Tivoli

″plus module″. This enables you to take advantage of the features that Tivoli provides for managing large enterprises, including monitoring Component Broker events and launching Component Broker tasks.

**“Event Monitoring” on page 8**
Monitoring and notification of significant events; actions taken by CBConnector System Management, changes to the state of your servers and applications, and other significant events that you subscribe to.

**“System Management Representation of Your Enterprise” on page 37**
CBConnector System Management represents everything that it can manage by a standard set of objects that you can act on in the same way. The objects are organized into a network that represents your enterprise, the CBConnector System Management administrative definition of it, and the CBConnector System Management components used to manage it.

**Related Concepts**

“Components used for System Management” on page 9
“System Management Representation of Your Enterprise” on page 37

## Installation of Software

There are two parts to installing software for Component Broker:

1. Installation of the base Component Broker software.

   This uses the Component Broker installation tool, with its own user interface. It adds the software needed by Component Broker to implement its system management (and other) functions. During the installation process, you can select various options to control which system management component software you want on a host.

   This is described in the *Quick Beginnings* and the *Planning, Performance, and Installation Guide*.

2. Installation of Component Broker application software.

   Each Component Broker application provides its own installation tool, with its own user interface. The tool adds the software needed by the application on managed hosts. This includes adding details of objects that CBConnector System Management is to use to represent the application, its DLLs, classes, and other objects.

   Application installation and configuration is normally described in information provided with the application family package. If you install an application onto the System Manager host, the System Manager can automatically distribute the application files to hosts where the application is configured to run.

   Component Broker provides a sample application installation tool for program developers to use as the basis for developing their own installation tools.

Both tools create objects that are used by Component Broker for system management.

**Related Concepts**

“Installing Applications” on page 222
“Administration of Enterprises” on page 6

## Administration of Enterprises

The hosts, servers, clients, and applications in your enterprise are there to serve your business. For effective system management, you are likely to want to organize them according to your business needs, rather than by their physical location. To do this, you can use CBConnector System Management to *administer* your enterprise, to do the following administration functions:

- Define your enterprise as one or more *management zones* that you want to administer as units. A management zone can define all or part of your enterprise and represents a *model* of what you want your enterprise to be like.
- Define alternative *configurations* of your management zones to provide support for changes in your business needs.
- In each configuration, define the things to be managed.
- Configure server groups for workload management.
- Verify that your configurations are valid.

CBConnector System Management uses a *system manager* (sometimes called an *SM Application* or *SMAppl*) to organize the administration of your enterprise and to coordinate your actions to operate servers and applications on hosts. The system manager stores all the *model objects* that you use for administration.

**Related Concepts**

"The Model World" on page 39
"The System Manager" on page 10
"Installation of Software" on page 5
"Operating your Enterprise"

## Operating your Enterprise

You can use CBConnector System Management to do the following operation functions:

- *Activate* configurations, to start all the servers, clients, and applications in the configurations
- Start and stop servers and applications
- Display and act on the status of servers and applications
- Display and change the attributes and relationships of servers and applications
- Subscribe to and react to events

These types of actions operate your servers and applications directly on their hosts.

CBConnector System Management uses an *agent* running on each managed host to control the servers and applications on the host. The agent runs separately from the managed servers, coordinates all actions on servers and applications on its node, and stores information about them.

**Related Concepts**

"The Image World" on page 40
"Installation of Software" on page 5
"Administration of Enterprises"

**Related Tasks**

“Chapter 15. Operate your Enterprise” on page 413

## Centralized Configuration of System Management Objects

The things in your enterprise that are to be managed by Component Broker are represented by a network of system management objects.

To administer your enterprise, you use the System Manager user interface to create a network of these objects, according to the Component Broker system management object model. The System Manager can verify the objects that you create, automatically create other objects that are needed, and can activate an entire network at one go.

When creating your network of system management objects, you can configure those objects to match your requirements. You can customize the attributes of system management objects, and their relationships with other such objects. For example, you can define your own values for attributes that control the running of an application server, and can change the host on which the server is to run by changing its *Configured Host* relationship from one host to another.

The System Manager stores the configuration data centrally and manages copies of that data where needed on managed hosts around your enterprise. Therefore, you can use the System Manager user interface as a single-point-of-control to centrally create all the system management objects for your enterprise, and to centrally operate your enterprise.

To operate your enterprise, you can use the same configuration process to change the central configuration data, and let the System System Manager automatically update your network of system management objects as needed. You can subscribe centrally to events on system management objects; for example, you can configure an application server so that you are notified if one of its attributes reaches a critical value.

**Related Concepts**

“Administration of Enterprises” on page 6
“Operating your Enterprise” on page 6

## Monitoring the Health of Application Servers

CBConnector System Management monitors automatically the health of application servers that it manages and can display the health of a server to give you an indication of how loaded the server is and whether or not it is short of resources. For example, for the server **My Server 1** running with good health, the status bar of the System Manager user interface would display:

My Server 1: runOnRequest running 3 3 **good**

The health values are used as a relative comparison of the health of several servers. For example, a server with "excellent" health is a relatively better target for application work than a server with "good" health. Two servers with "poor" health are equally the poorest target for application work.

The health values can also be used as an input to workload management decisions.

### Related Concepts

“Application Servers and Server Groups” on page 25
“Workload Management” on page 335

## Event Monitoring

CBConnector System Management monitors your enterprise for the following types of events:

- Actions that you told CBConnector System Management to perform on your servers and applications; for example, to start a server. These events are monitored by default.
- Events caused by changes in your servers and applications other than by CBConnector System Management actions; for example, the state of a server changes from **Excellent** to **OK**. You must *subscribe* to these events to tell CBConnector System Management to monitor them.

When an event occurs, CBConnector System Management records the event and can display an Event Monitor window to inform you about the event. You can use the Event Monitor window to open the object affected, to enable you to react to the event.

### Related Concepts

“Operating your Enterprise” on page 6
“Event Subscriptions” on page 443

### Related Tasks

“Monitor Events” on page 419
Display The Events Log (page 420)
React to Events (page 420)
Remove an Entry from the Log File (page 420)
Create or Change Event Subscriptions (page 420)
Delete Event Subscriptions (page 421)
“Display Information in Component Broker Logs” on page 423

## Integrated System Management with Tivoli

The Tivoli system management product provides a number of features for managing large enterprises. Component Broker system management is integrated with Tivoli through a Tivoli ″plus module″, which provides the following the main types of support:

- A set of plus module icons in the TivoliPlus window that can be used to launch Component Broker system management tasks; for example, to start the System Manager user interface
- Events added to a Component Broker error log on a managed host can be sent to Tivoli, and so can be monitored using the Tivoli Event Console monitoring facilities.

To enable Component Broker system management integration with Tivoli, you complete the following tasks:

1. “Install and Configure the Component Broker plus module for Tivoli” on page 47 to add icons to Tivoli, enable its event server to recognize Component Broker events, and define the Tivoli event consoles that are to monitor those events.
2. “configure and enable Tivoli event monitoring” on page 50 so that Component Broker sends events to the Tivoli event server whenever an entry is added to a Component Broker error log.
3. (Optional) If you want Tivoli to be able to launch the Component Broker System Manager user interface, you need to “Install a System Manager User Interface” on page 51 on the Tivoli host.

**Related Tasks**

“Integrate Component Broker System Management with Tivoli” on page 47

## Components used for System Management

This topic describes the components used for Component Broker System Management. It contains information about the following:

- “The System Manager” on page 10
- “The SM Agents” on page 11
- “System Manager User Interface” on page 12
- “Common Data Model” on page 11
- “Configuration Data” on page 11
- “ORB Daemon” on page 14
- “Log files” on page 14

The host on which the System Manager runs, along with all the hosts that it manages, is referred to as the *Component Broker network*, as shown in Figure 2 on page 10. The Component Broker network can be part or all of your systems enterprise. The System Manager interacts with the objects on the managed hosts in its network through an *SM Agent* that runs on each managed host.

If your enterprise is divided up into two or more Component Broker networks, each network is a separate entity, managed by its own System Manager, and with its own system name tree. Note that you do not need to create separate Component Broker networks in order to partition the hosts in your enterprise. You can group your managed hosts into separate *workgroups* within the same Component Broker network.

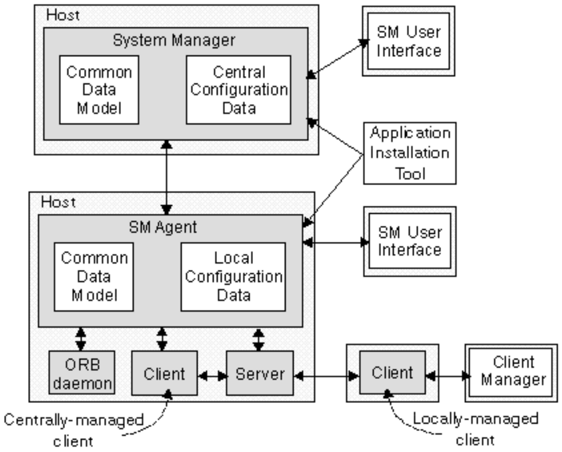


*Figure 2. A Component Broker Network*

The System Management components cooperate with other prerequisite components for Component Broker (for example, DCE clients and servers) to provide the system management functions for your enterprise.

## The System Manager

The CBConnector *System Manager* provides the logic that manages a Component Broker network. The System Manager interacts with the objects on the managed hosts in its network through an *SM Agent* that runs on each managed host.

The System Manager relates the definition of objects within its Component Broker network (the model world data stored in its *central configuration data*) to the objects within the real enterprise (the image world and install world data stored in *local configuration data* by the SM Agents).

It presents this data to users through the *System Manager user interface*. Through the user interface, you can operate servers, clients, and applications on the managed hosts and can administer those objects in management zones defined in the central configuration data. The structure of the data in the central configuration data is defined by the System Manager's copy of the *Common Data Model*.

A System Manager has its own process that runs as a Windows NT service or AIX resource.

The System Manager is sometimes referred to as the "System Management Application" or "SM Application".

### Related Concepts

"System Management Representation of Your Enterprise" on page 37
"The SM Agents" on page 11
"System Manager User Interface" on page 12
"Common Data Model" on page 11
"Configuration Data" on page 11

## The SM Agents

There is one *SM Agent* on each centrally-managed host. The SM Agent is the means by which the System Manager communicates with the objects to be managed on that host. It interacts directly with those objects, passing data back to the System Manager and acting on the objects on behalf of the System Manager.

An SM Agent contains *local configuration data*. This is a full definition of the objects to be managed on the agent's host (the *image world*) and a full definition of the applications installed on that host (the *install world*). The SM Agent creates the image world by expanding a slave copy of the part of the model world that is relevant to the SM Agent's host and combining that with information taken from the install world.

An SM Agent has its own process that runs, as an NT service or AIX resource, separate from the servers that it controls.

Generally, system management actions on an SM Agent are made through the user interface connected to the System Manager. However, you can attach the user interface directly to an SM Agent; for example, as a local debug aid.

**Related Concepts**

"System Management Representation of Your Enterprise" on page 37
"The Model World" on page 39
"The System Manager" on page 10
"Common Data Model"
"Configuration Data"
"Operating CBConnector System Management Components" on page 431

## Common Data Model

The *common data model* is a template that describes the structure of "Configuration Data" within CBConnector System Management. It comprises folders of objects that form a hierarchical tree structure. The model describes:

- The folders that can exist
- The attribute names, types, limits, and default values of objects in those folders
- The relationships that can exist between objects

The common data model is stored in the system manager and SM agents. These components use the common data model to organize their configuration data.

**Related Concepts**

"System Management Representation of Your Enterprise" on page 37
"The System Manager" on page 10
"The SM Agents"
"Configuration Data"

## Configuration Data

*Configuration data* defines the topology of an enterprise. Its structure is defined by the "Common Data Model" and represents an instance of that data model. The configuration data is held in the system manager and the agents.

- The *central configuration data* held in the system manager defines the high level of an enterprise’s topology (as the model world of management zones, configurations, hosts, and so on).
- The *local configuration data* held in an SM agent defines the low level of an enterprise’s topology. This local data is based on a copy of the central configuration data that is specific to the agent’s host (the image world). It is combined with install world data (for applications installed on the host) to produce a full configuration definition for that host.

### Related Concepts

“The System Manager” on page 10
“The SM Agents” on page 11
“Common Data Model” on page 11

### Related Tasks

“Chapter 15. Operate your Enterprise” on page 413

## System Manager User Interface

The Component Broker System Manager user interface has standard features familiar to users of other user interfaces, as well as other features to simplify your system management tasks:

- Standard window layouts and control features
- representation of your enterprise
- An object editor to display and change object attributes
- A history list of objects visited in the current session with the user interface
- A hotlist of significant objects preserved across user interface sessions
- Subscription to events in your enterprise
- Online help information

For example, the main window of the System Manager user interface, called the **Information Controller**, is shown in Figure 3.

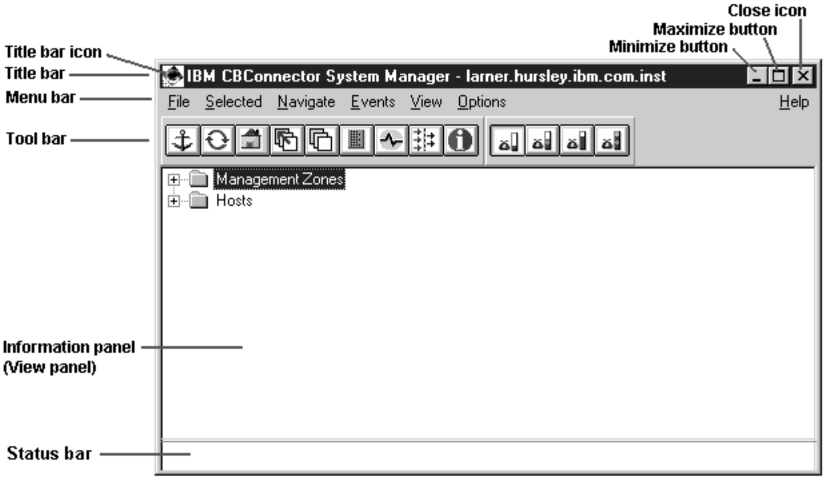


*Figure 3. The Information Controller window.*
*Showing the Home view of the CBConnector System Management object network.*

The System Manager user interface is effectively the same on all platforms supported by Component Broker. Most images of the user interface that are provided in these online topics were captured from the System Manager user interface on Windows NT. There are some minor differences with other platforms caused by how some window functions need to be implemented. Any significant differences are described in these online topics.

The System Manager user interface is normally connected to a System Manager to form a **single point of control (SPOC)** for all managed hosts in an enterprise. There can be several instances of the user interface to provide different SPOCs for system management. The user interface can also be connected directly to an SM Agent on a host to provide a system management interface for managing only that host.

For more information about the System Manager user interface, see “Appendix A. Features of the User Interface” on page 433. For details about the windows and functions of the System Manager user interface, see the context-sensitive F1 help provided by the user interface.

**Related Concepts**

“Appendix A. Features of the User Interface” on page 433
“The Object Editor Window” on page 439
“Session History” on page 441
“Hotlist” on page 442
“Event Subscriptions” on page 443
“User-Level Settings and Object-Level Filters” on page 438
“Help Information” on page 449
“File Browser” on page 449
“Chapter 2. Use the System Manager User Interface” on page 55

**Related Tasks**

“Start the System Manager User Interface on Windows NT” on page 56
“Start the System Manager User Interface on AIX” on page 58
“Exit the User Interface” on page 60
“Chapter 2. Use the System Manager User Interface” on page 55

## DDL Files for Applications

The installation package for an application family contains a **DDL file** that describes the contents of the application family. It describes the applications in the family and the objects, attributes, and relationships that make up each application. For example, the DDL file for an application family defines the following:

- The applications to run on servers and their relationships to objects that they provide
- The applications to run on clients and their relationships to objects that they provide
- The classes, DLLs, homes, containers, and other objects provided by the applications, and appropriate relationships between such objects
- Appropriate attributes of the applications and other objects in the application family

The application family installation program uses the information in the DDL file to create *Install objects* that the System Manager can use to define and configure the applications.

When you use Object Builder to create an application family, it generates a DDL file for the application family. Before you generate the install image for an application family package, you can add other objects to the DDL file.

You do not normally change DDL files after the application family has been installed into Component Broker. When you load an application family into Component Broker, each application in the DDL file is represented as an **available application** through the System Manager user interface. If you need to customize the application within Component Broker, you normally do so by changing model objects for the application through the System Manager user interface.

## ORB Daemon

CBConnector System Management manages *ORB daemons* through SM agents. Each server host has a *Daemon Image* that represents the ORB daemon on that host and through which the daemon can be configured.

## Log files

All CBConnector components produce error, activity, and trace data that is saved in *log files*. The *File browser* can be used to view log files.

### Related Concepts

Activity log (page 415)
Error log (page 415)
Trace log (page 416)
“File Browser” on page 449
“Sources of Information” on page 415
“Problem Determination” on page 471

### Related Tasks

“Display Information in Component Broker Logs” on page 423

## System Management Samples

When you install a System Manager, CBConnector automatically creates a Management Zone called *Sample Cell and Work Group Zone*, which contains a sample configuration for a standard basic set-up. The Configuration contains models for a sample cell and work group name tree, and a TCP/IP protocol. It also contains all the required relationships between those models and the Host model. If you activate this configuration, the host is configured as the central name server for the sample cell and work group.

If you select to install the sample applications, then CBConnector automatically creates a Management Zone called *Sample Application Zone*. This zone contains a sample Configuration which, when activated, creates a server configuration ready for the sample Insurance application that is shipped with CBConnector. After activating the sample Configuration, you can try the sample application immediately.

In the Sample Application Zone, the Sample Configuration contains a sample Server (free standing) called *Sample Server 1* and a sample Client Style called *Sample Client Style*.

If you install other hosts as server hosts (with an SM Agent), then CBConnector creates other sample Servers (free standing) called *Sample Server n* in the Sample Configuration. For each new server host, the sample Server Model uses an increasing integer *n*). Also each server host is configured to use the cell and work group nametree server on the manager host and to use the manager host as the boot strap host.

**Related Tasks**

"Example: Use the Sample Configuration for a Single-Host Environment" on page 95
"Example: Use the Sample Configuration for a Multi-Host Environment" on page 126

## Parts of your Enterprise Managed by Component Broker

This topic provides general information about the parts of your enterprise that are managed by Component Broker. It describes the main concepts and terms used for parts of your enterprise.

Your enterprise can be considered as two environments, the *host environment* and the *application environment*, as shown in the following figure.

**Host environment**
This comprises the System Manager host, all server and client-only hosts that it manages, the hosts' name servers and protocols, the Component Broker cell and work groups, and any remote name contexts linking to other system name trees.

**Application environment**
This comprises your applications, the application servers and client styles that applications run on, and application resources like DLLs, Homes, and connections to tier-3 systems.

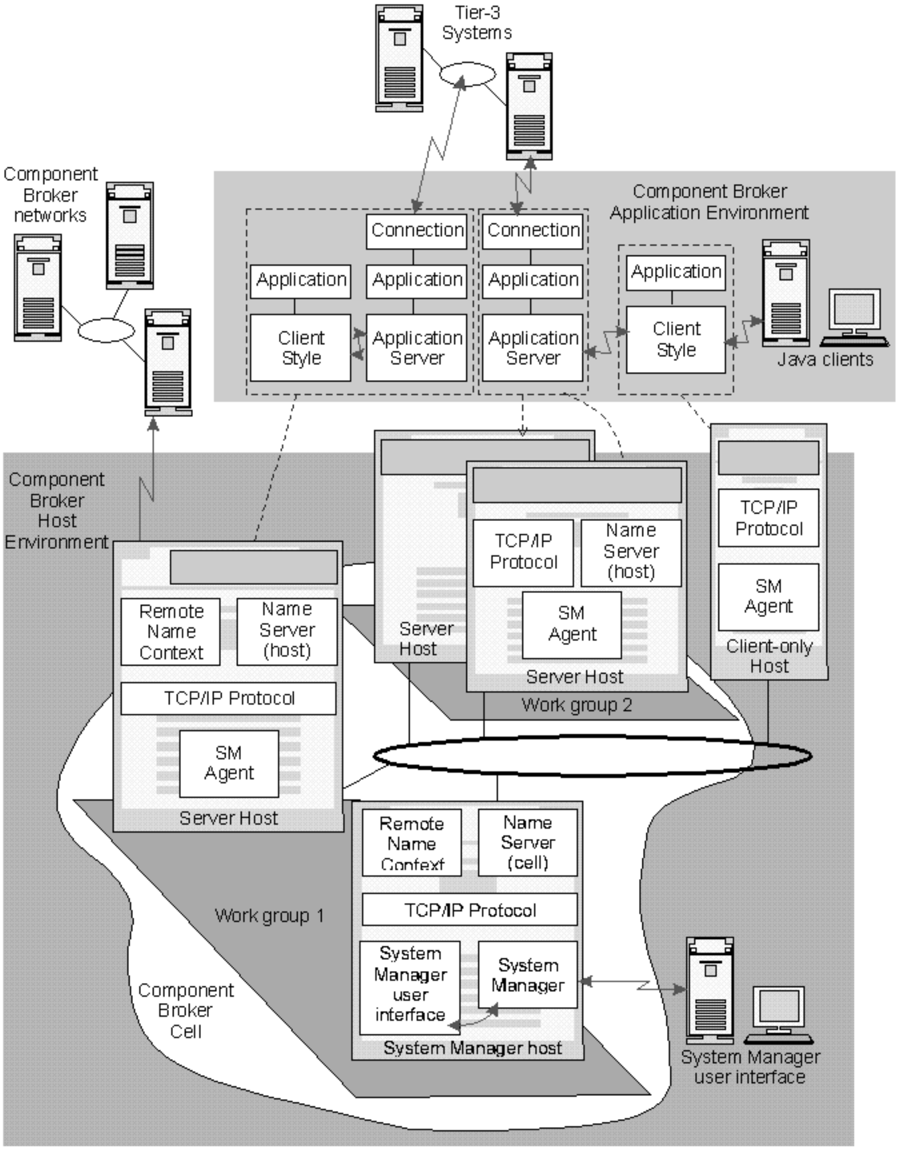


*Figure 4. A Component Broker Enterprise*

You should define your host environment in one management zone, and define your application environment in one or more *other* management zones. Separating the host environment and application environment in this way optimizes the configuration of your enterprise and the ease with which you can administer it. Within each management zone you define one or more alternative *Configurations*. For more information about management zones and Configurations, see the following topics:

**“Management Zones” on page 17**
All or part of your enterprise that is to be managed as a single unit.

**“Configurations” on page 18**
Alternative implementations of a management zone.

For more information about what you should do to configure parts of your enterprise, see “Chapter 1. An Overview of Administering Component Broker Enterprises” on page 1.

For more information about the host environment, see the following topics:

**“Hosts” on page 20**
Host computers on which your managed clients and hosts run, and on which you have installed Component Broker system management components.

**“Cells and Work Groups” on page 19**
Groups of managed hosts to be managed as multi-host units.

**“The System Name Tree” on page 21**
The system name tree used by Component Broker and links to the name trees of other Component Broker enterprises.

For more information about the application environment, see the following topics:

**“Application Servers and Server Groups” on page 25**
Servers, and groups of servers, that run on managed hosts to support your applications.

**“Client Styles” on page 26**
Types of clients that request operations on objects managed by application servers.

**“Applications” on page 27**
The applications running on clients and servers that provide your business services.

**“Database Aliases” on page 28**
Configuration data about relational databases used by applications.

**“Connections to Tier-3 Systems” on page 29**
APPC, ECI, and HOD connections between Component Broker application servers and tier-3 systems like CICS and IMS.

**Related Tasks**

“Chapter 3. Create a New Single-Host Enterprise” on page 85
“Chapter 4. Create a New Multi-Host Enterprise” on page 101
“Chapter 5. Configure a new Application Environment” on page 133

## Management Zones

A *management zone* defines a part of an enterprise that is to be managed as a unit.

You configure the hosts in your Component Broker network in one management zone, to group the hosts into the DCE cell used by Component Broker and into one or more work groups. This network management zone is used to create and update the name tree used by Component Broker.

You are recommended to configure your application environment into one or more other management zones separate from your network management zone. For example, you may have several application management zones to manage your enterprise by separate geographical groupings, by business application groupings, or to separate development servers from production servers. Each application management zone represents a separate logical grouping of applications and the servers and clients that they run on. Several application management zones can make use of the same hosts in your network, or can use completely separate groups of hosts within your network.

A management zone comprises one or more alternative *“Configurations”* . Each configuration groups the related servers, clients, and applications that are to be running for a specific implementation of the management zone. Typically, a management zone may have several configurations; for example, to respond to regular changes in business needs.

By dividing an enterprise into management zones, an administrator can alter the configuration and characteristics of one management zone without affecting the functioning of other management zones.

You should use one management zone for your host environment (your hosts, the Component Broker cell, workgroups, name servers, and host protocols) and one or more other management zones for your application environment.

You are recommended to keep your application management zones small This makes it easier to manage each management zone and makes your enterprise more flexible.

**Related Concepts**

“Configurations”
“Cells and Work Groups” on page 19
“Application Servers and Server Groups” on page 25

**Related Tasks**

“Create a new Management Zone and Configuration for your Host Network” on page 110
“Create a new Management Zone and Configuration for your Application Environment” on page 139
“Chapter 10. Administer Management Zones and Configurations” on page 253

## Configurations

A *Configuration* contains definitions of servers, clients, and applications that you want to manage as a unit. It is an implementation of a Management Zone. Typically, a Management Zone may have several Configurations; for example, to respond to regular changes in business needs. For example, an administrator may define a Management Zone to group the servers and applications for a business area, and within that Management Zone define a ″day″ Configuration and a ″night″ Configuration. The ″day″ Configuration groups the objects needed for the workload during the daytime, and the ″night″ Configuration groups the objects needed for the workload overnight. (The day and night Configurations may have some servers and applications in common.)

Within a Configuration, each server that can be started is defined by a *Server* model. The applications that can be used by those servers are defined by *Applications* that have been configured onto the Servers.

You can use a **Verify** action on a Configuration to check that all the objects it contains are properly defined and have the relationships and attribute values needed for the Configuration to be activated. This action attempts to establish any missing relationships that it can automatically, and warns you of any things to correct before you activate the Configuration.

You can use a single **Activate** action on a Configuration to start all its servers and applications on their hosts.

**Related Concepts**

"Management Zones" on page 17
"Hosts" on page 20
"Application Servers and Server Groups" on page 25

**Related Tasks**

"Add a new Configuration to a Management Zone for your Application Environment" on page 254
"Verify a Configuration" on page 255
"Activate a Configuration" on page 256

## Cells and Work Groups

In an enterprise of several host computers the hosts are often logically related, typically by the business or geographical area that they support. To manage several hosts better as multi-host units, you can group them into *cells* and *work groups*. CBConnector System Management applies no restrictions on the hosts in either type of group; the hosts are only grouped logically to aid system management.

Each such group comprises a number of hosts on which servers run to support business applications. On one of the hosts in the group there is also a server that provides a dedicated naming service for the group. That *naming server* resolves all requests by name for hosts, servers, and other objects in the group.

Hosts in a cell can also be members of a work group. Therefore, you can have an enterprise that comprises a number of cells that group hosts similar to Windows NT or DCE cells for general administration and several workgroups that group the same hosts into business areas for administration along business lines.

For example, an enterprise of six hosts numbered 1 through 6 is shown in Figure 5 on page 20. Hosts 1 and 2 support a central stock application. Hosts 3 through 6 support a branch sales application. Hosts 1 and 2 are members of one cell and the other hosts are members of a second cell. Hosts 1, 3, and 4 are members of one work group and hosts 2, 5, and 6 are members of a second work group.

There is a one-to-one relationship between a DCE Cell and a Component Broker Network. In this context, a Component Broker Network is considered to be all of the managed server hosts *which are managed by the same System Manager*. Therefore, all of the managed server hosts that use the same System Manager must belong to the same DCE Cell.

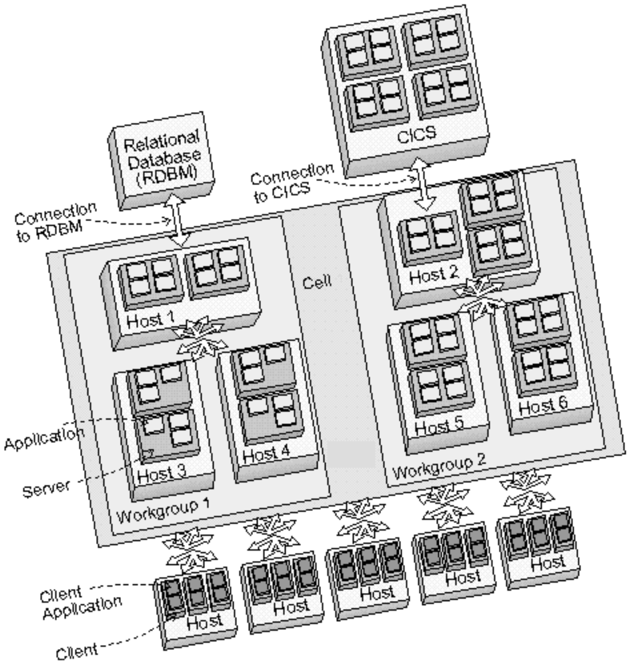


*Figure 5. Cells and Work Groups, an Example Enterprise*

**Related Concepts**

"Management Zones" on page 17
"Hosts"
"The System Name Tree" on page 21

**Related Tasks**

"Configure the Component Broker Cell and Minimum Work Group" on page 111
"Configure a new Work Group" on page 115
"Configure a Host as a Member of Non-preferred Work Groups" on page 118

## Hosts

Component Broker represents each host computer that it is to manage by a *Host* object within the Component Broker network displayed by the System Manager user interface.

Component Broker automatically creates a Host object (and assigns its name) if you install a System Manager or SM agent on that host.

You can use a Host object to configure objects that are to be used on the host computer; for example, you can configure servers to run on that host.

Hosts can be grouped arbitrarily into *cells* and *workgroups*.

Component Broker also automatically creates a *Host Image* for each host computer. You can use the Host Image to display and act on the runtime view of the host computer; for example, to display the health of servers running on that host.

**Related Concepts**

“Cells and Work Groups” on page 19
“The SM Agents” on page 11
“System Management Representation of Your Enterprise” on page 37

**Related Tasks**

“Configure a new Host into your Network” on page 117

## The System Name Tree

This topic outlines the structure of the system name tree in a Component Broker environment. (This name tree is generally referred to as the *Component Broker Name Tree*.) It should help you understand the configuration choices that you make when configuring parts of the Name Tree.

For general information about the Naming Service, including descriptions of the system name space, the various name trees, and visibility of objects, see the ″Naming Service″ chapter in the *Advanced Programming Guide*.

Each Component Broker network (all the hosts managed by the same System Manager) has its own system name tree, implemented by a name server running on each managed server host.

The host name server on a managed server host is called ″<hostname> Name Server″, and is created automatically by the System Manager whenever any of the following factors are true:

- There is an application server configured on the host
- The host is the bootstrap host for a managed client
- The host is a serving host for a cell or a workgroup
- The host is member host of a cell or a workgroup
- The host is a preferring host of a cell or a workgroup
- The host is designated as the managing host for a server group

The name server provides the naming service for the host, and in some cases for a cell and workgroup. The name server is used by Component Broker to house a number of system objects, including naming contexts used in the system name space, factory finders, location objects, event channels, and so on. Client-only hosts can be configured without a naming server, to use the name server on another *bootstrap host*.

Name servers are controlled automatically by the System Manager. They cannot be stopped while related application servers are still running, and cannot be deleted, renamed, moved, or have applications configured on them.

The System Manager creates the name server, and the **Name Server Image** that it uses to represent that server, during the first activation of any Configuration where at least one of the above factors is true. The Name Server Image that is created contains one of the following applications:

- iHostServices
- iWorkGroupServices
- iCellServices

These applications define Name Tree Image (NTI) objects in the Name Server Image. The iHostServices application defines a Host NTI object. The iWorkGroupServices application defines a Host NTI object and a WorkGroup NTI object. Lastly, the iCellServices application defines a Host NTI object, a WorkGroup NTI object and a Cell NTI object. The responsibilities of these NTI objects are as follows:

- Cell NTI - builds and manages the cell name tree
- WorkGroup NTI - builds and manages a workgroup name tree
- Host NTI - builds and manages the local root context for the host, the host name tree, links to the preferred cell and workgroup, and links back to its host tree in the hosts context of the cell and workgroup trees of which it is a member.

These NTI objects only build and maintain the basic structure of the name tree. For example, for factory finders the Cell NTI creates its root context and bind in resources/factory-finders. However, the binding of the default cell factory finder as resources/factory-finders/cell-scope is done by the factory finder itself, not by the Cell NTI object. Likewise, the Cell NTI binds resources/factories, but resources/factories/somlcRepository is bound in by the LifeCycle service.

A name server always has only one Host NTI. Because there is only one cell in a Component Broker network, there is only one Cell NTI, in only one name server. Workgroup NTIs can exist in any number within name servers defined with the iWorkGroupServices application or iCell Services application. How this is managed from a configuration perspective is explained in “Name Tree Configuration Management Rules” on page 168.

If you want to prevent unauthorized access to the Name Server and the resources it houses, you should “Configure Security for a Server” on page 323, as you would for your own application servers. If you will be accessing the Name Server from Java clients over the SSL-based authentication facility, you must provide a certificate for this server and include its trust-basis in the client keyrings for any clients that will access it.

The system name tree for a Component Broker network can be linked to other name trees to enable objects in the network to talk to objects outside that network. Separate name trees can be linked by *remote name contexts*. For example, remote name contexts can be used to link the system name tree for one Component Broker network on Windows NT with the system name trees for other Component Broker networks on S/390, Windows NT, and AIX. This also enables the system name tree for a Component Broker network to be linked with the name tree for a non-Component Broker host that has an OMG-compliant implementation of CosNaming and that supports the bootstrapping protocol. “Cells and Work Groups” on page 19

**Related Concepts**

“Administration of Enterprises” on page 6
“Hosts” on page 20
“Cells and Work Groups” on page 19
“Remote Name Context Connections to other System Name Trees” on page 23
“The Component Broker Name Tree and DCE” on page 23
“Name Tree Configuration Management Rules” on page 168

**Related Tasks**

“Create a new Management Zone and Configuration for your Host Network” on page 110

page 110
“Configure the Component Broker Cell and Minimum Work Group” on page 111
“Configure a new Work Group” on page 115
“Configure a new Host into your Network” on page 117
“Verify that a Name Server is in the DCE CDS” on page 119
“Chapter 6. Administer your Host Environment” on page 167

## Remote Name Context Connections to other System Name Trees

You can use *remote name contexts* to link the system name tree for one Component Broker network to the name tree for another Component Broker network or a non-Component Broker host. This enables objects in the Component Broker network to talk to objects outside that network.

Component Broker system management defines and manages such contexts by *Remote Name Context* objects. The context specified in a Remote Name Context is bound into either the system name tree for the local Component Broker network or the remote name tree depending on the binding direction specified in the Remote Name Context.

Remote Name Contexts enable separate system name trees to contain references to each other. For example, this enables the system name tree for one Component Broker network on Windows NT to be linked with the system name trees for other Component Broker networks on S/390, Windows NT, and AIX. It also enables the system name tree for a Component Broker network to be linked with the name tree for a non-Component Broker host that has an OMG-compliant implementation of CosNaming and supports the bootstrapping protocol.

### Related Concepts

“The System Name Tree” on page 21
″Naming Contexts″ in the *Advanced Programming Guide*
″Object Names″ in the *Advanced Programming Guide*

### Related Tasks

“Configure a Remote Name Context for Use on Hosts” on page 172

## The Component Broker Name Tree and DCE

This topic outlines the relationship between the Component Broker name tree and DCE. It provides the following information about the name tree:

- **Using DCE Director to examine the Name Tree (page 23)**
  What the DCE Director can show you about the bindings for Component Broker, and how to understand those bindings.
- **Relationships Affecting Name Tree Structure (page 24)**
  Relationships between hosts and cells and workgroups (defined to the Component Broker System Manager) that affect the structure of the Name Tree.
- **“Name Tree Configuration Management Rules” on page 168**
  Rules that govern how the name tree is configured and the changes that can be made.

### Using DCE Director to examine the Name Tree

Because all name tree bindings are stored in the DCE CDS, you can use the DCE Director to inspect the Component Broker name tree.

There are the following three bindings for Component Broker in the DCE CDS Root Directory:

**CBC-root**
The cell name tree context.

**CBC-workgroup-roots**
Each workgroup name tree context that is bound into this context.

**CBC-local-roots**
The local root context of each server host that is bound in this context. Bound below each local root context is the host name tree context for that host.

When using the DCE Director, you will notice that all the names either end in a period (″.″) or contain a period. This period is inserted by the Component Broker naming service when binding the name into the DCE CDS. OMG-defined names contain two parts: an ″id″ and a ″kind″. Most names only have an id and are stored in DCE as **id.** (note the trailing period), while those with a kind are stored as **id.kind** (with no trailing period).

In general, the period is not used in documentation when referring to an id-only name. Therefore, a name in the DCE CDS, which is shown as **host./resources./factory-finder.** is generally documented as **host/resources/factory-finder** (without the trailing periods).

### Relationships Affecting Name Tree Structure

The System Manager allows you to define relationships between hosts and cells and workgroups. These relationships affect the structure of the System Name Tree that is built, as follows:

- **Preferred Hosts to Preferred Cell/Workgroup**

  Both Cell and Workgroup models can have Preferring Hosts relationships. When viewed from the point-of-view of the Host model, these relationships are called ″Preferred Cell″ and ″Preferred Workgroup″. The result of these relationships is a link from the local root for the host to the cell or workgroup tree.

  From the local root context of host<hostname>:

  – Preferred Cell
    **/.:** links to the preferred cell for host <hostname>.

  – Preferred Workgroup
    **/workgroup** links to the preferred workgroup for host <hostname>.

- **Member Hosts to Member of Cell/Workgroup**

  Both Cell and Workgroup models can have Member Host relationships. When viewed from the point-of-view of the Host model, these relationships are called ″Member of Cell″ and ″Member of Workgroup″. The result of these relationships is a link within the host context of the cell or workgroup tree back to the host tree.

  From the local root context of any host:

  – Member of Cell
    **/.:/host/<hostname>**links to the host tree for <hostname>.

  - Member of Workgroup **/.:/workgroup/<workgroupname>/hosts/<hostname>**links to the host tree for <hostname>.
- **Serving Host to Served Cell/Workgroup**

  Both the Cell and Workgroup models need to have a Serving Host relationship. When viewed from the point-of-view of the Host model, these relationships are called ″Served Cells″ and ″Served Workgroups″. These relationships define the host whose name server is responsible for the run-time management of the cell or workgroup name tree.

#### Related Concepts

"The System Name Tree" on page 21
"Name Tree Configuration Management Rules" on page 168
"Cells and Work Groups" on page 19
"Management Zones" on page 17
"Hosts" on page 20

#### Related Tasks

"Create a new Management Zone and Configuration for your Host Network" on page 110
"Configure the Component Broker Cell and Minimum Work Group" on page 111
"Configure a new Work Group" on page 115
"Configure a new Host into your Network" on page 117
"Verify that a Name Server is in the DCE CDS" on page 119
"Chapter 6. Administer your Host Environment" on page 167

## Application Servers and Server Groups

An enterprise managed by CBConnector System Management comprises a number of *application servers* running on one or more hosts.

In the Component Broker programming model a *client application* requests operations on one or more objects managed by an application server.

In the simplest case, you can have one free standing application server running on only one host to support your applications.

Typically the use of just one free standing application server, confined by the resources of a single host computer, does not have the capacity to support the work generated by the many clients wanting to use a particular application. As more and more users start using a client application the workload on the associated application server increases and response times suffer. Using just one application server also raises an availability issue as you have a single point of failure in your enterprise where a problem would prevent use of an entire application. For these reasons, in a typical enterprise, the administrator normally configures an application onto many application servers. To make this a relatively straightforward administrative task Component Broker system management uses the concept of a *server group*. Servers in a server group are typically distributed across many hosts in the network.

An application server is defined within a Configuration of one of your application Management Zones by either a *Server (member of group)* or *Server (freestanding)*:

- A free standing application server generally has unique characteristics and supports a unique set of applications. Such a server is defined by a *Server (free* *standing)*. The applications supported by the server are defined by Applications configured onto the Server (freestanding). When the Configuration containing the Server (freestanding) is activated, the System Manager builds a corresponding Server Image, to represent the server in the SM Agent on the target host, and starts the application server on that host. It also creates Images for applications and other objects configured onto the Server (freestanding).

- In a Configuration that uses several application servers of the same style, it is usual to define the style of server by a *Server Group* and within that object a *Server (member of group)* for each application server in the group. The applications supported by those servers are defined by Applications configured onto the Server Group.

  The Servers (member of group) generally have the same characteristics and support the same applications. However, you can change a few attributes of individual Servers (member of group), to tailor their characteristics and function.

  You can configure a Server Group as a *controlled server group* to enable workload management across its servers.

  When a configuration containing a Server Group is activated, its Servers (member of group) are used to build and start the application servers on their target hosts. The activate action creates a Server Image (in the SM Agent on the target host) to represent each server. It also creates Images for applications and other objects configured onto the Server Group.

You can use an action to convert a Server (free standing) into Server (member of group) or the other way around.

**Related Concepts**

"Workload Management" on page 335
"Controlled Server Groups" on page 336
"Server Group Control Point (SGCP)" on page 337
"Server Group Gateway (SGGW)" on page 338

**Related Tasks**

"Define and Configure Servers and Server Groups" on page 182

## Client Styles

CBConnector System Management can be used to manage types of clients (*Client Styles*) and applications that run on them (*Client Applications*).

For system management, there are the following categories of client:

- Centrally-managed clients
- Centrally-configured clients
- Non-managed clients

*Managed clients* run on host computers onto which CBConnector System Management can install appropriate management software. Non-managed clients typically run on hosts onto which management software cannot be installed (for example, the Java-applet type of client) so direct management is not possible.

However, you can centrally control the configuration of Java clients through the use of Java properties files. Such clients are referred to as *configured clients*.

Clients can be installed with or without system management capability. For managed clients, the usual method of management is centrally by the System Manager user interface running on a System Manager host communicating with the SM Agent on the client host (*centrally-configured clients*).

The main types of clients supported by Component Broker are as follows:

- Java Application Clients
- Java Applet Clients
- C++ Clients
- Configured Clients

For more information about the client types supported, see “Clients” on page 207.

**Related Tasks**

“Configure a new Client Style” on page 151
“Configure Applications onto a Client Style” on page 213
“Define and Configure Client Styles” on page 212
“Chapter 8. Administer Clients” on page 205

## Applications

Component Broker can manage *applications* running on clients and servers. It represents client-based applications as *Client Applications* and server-based applications as *Applications* (without any prefix).

Generally, an application installation tool is used to install application software onto a host. A related set of applications (programs) and their components (for example, DLLs and class libraries) are installed as an *application family*. When you install an application package, Component Broker automatically creates an *Application Family Install* object and, within that, *Application Install* objects or *Client Application Install* objects and related Install objects to represent components used by the applications; for example:

- Dynamic link libraries (DLLs)
- Interface repositories
- Managed object classes and Data Object classes
- Homes and Containers
- Profiles and Profile Classes
- Connections to tier-3 systems and databases

When an application is installed, its components are created automatically as Install objects (for example, **DLL Installs**), and related to the Application Install.

The Install objects often predefine all the details of the application family, so that you only need to complete some basic configuration tasks to be able to use the applications.

For each application installed, the System Manager also automatically creates an *Available Application* or *Available Client Application* object, as appropriate. The System Manager user interface displays the folders of these objects prominently so that you can easily see what applications are available to Component Broker.

You can create an *Application* within a Configuration of one of your application Management Zones to configure an application for use. An Application is created by dragging an Available Application and dropping it onto a Configuration. This automatically creates models for objects needed to configure and administer the application; for example, Database Aliases and ECI Connections.

An Application can be configured onto all appropriate Server Groups and Servers (freestanding) within the same Configuration, to define on which servers the application is to be available.

When the Configuration containing a configured Application is activated, the application is made available on the related servers. If the application software is not available on the hosts on which the servers are to run, the activation process copies the software needed to those hosts.

Similarly, you can drag an Available Client Application and drop it onto a Configuration to define and configure a client application for use by one or more client styles within that Configuration. When the Configuration containing a configured Client Application is activated, the application is made available on the related client styles.

**Related Concepts**

"Application Servers and Server Groups" on page 25

**Related Tasks**

"Add an Application into a Configuration of your Application Environment" on page 228
"Add a Client Application into a Configuration of your Application Environment" on page 230
"Load a new Application" on page 141

## Database Aliases

Many applications use a relational database to store their persistent data, and need administrators to provide the names of the databases to be used on their hosts. This is done by selecting and editing the objects provided in the Application Model's *Provides Database Alias* folder. This folder is created and populated automatically when the Application model is created from an Application Install object.

Detailed instructions about working with database aliases are provided with the associated applications.

**Related Concepts**

"Application Servers and Server Groups" on page 25

**Related Tasks**

"Add an Application into a Configuration of your Application Environment" on page 228
"Load a new Application" on page 141
"Configure a new Connection to a Database for use by Applications" on page 237

## Connections to Tier-3 Systems

Applications running on Component Broker servers can interact with applications running on tier-3 systems (for example, CICS for OS/390, CICS for Windows NT, or IMS systems). Within Component Broker, the mechanism for communication is defined by **Connection** objects. Each Connection object is used to configure and manage one or more connections to a logical tier-3 system, up to a maximum predefined on the Connection object.

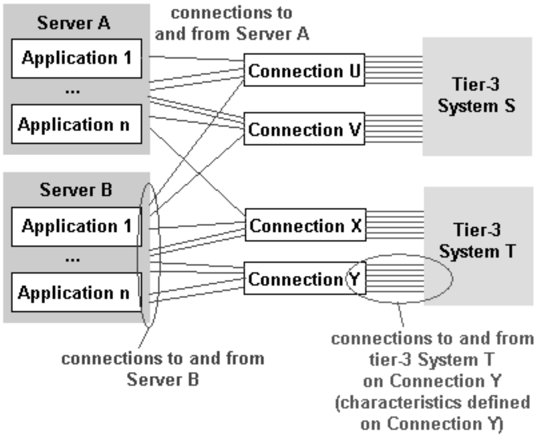

Server A
Application 1
...
Application n
connections to and from Server A
Connection U
Connection V
Tier-3 System S
Server B
Application 1
...
Application n
Connection X
Connection Y
Tier-3 System T
connections to and from Server B
connections to and from tier-3 System T on Connection Y (characteristics defined on Connection Y)

*Figure 6. Connections to Tier-3 Systems*

### Types of Connections

There can be the following types of Connections within Component Broker:

**"HOD Connections to Tier-3 Systems" on page 32**
A **HOD Connection** is used to configure and manage one or more connections to a TN3270 listener running on a tier-3 CICS region or IMS server. The connection uses a Java-based TN3270 client provided by Host On-Demand (HOD), which is shipped with Component Broker. Host On-Demand is a member of the eNetwork software family that is a Java-based solution that incorporates industry-standard Telnet 3270 (TN3270) protocols.

**"ECI Connections to Tier-3 Systems" on page 31**
An **ECI Connection** is used to configure and manage one or more connections through the CICS Transaction Gateway to a tier-3 CICS region. The connection uses the underlying External CICS Interface (ECI) communications provided by the CICS Transaction Gateway. The CICS Transaction Gateway is shipped with Component Broker.

**"APPC Connections to Tier-3 Systems" on page 33**
An **APPC Connection** is used to configure and manage one or more connections that use LU6.2 to communicate with tier-3 systems to update data as part of a Component Broker transaction.

The LU 6.2 connections are provided by an IBM Communications Server running on the same host as Component Broker application server.

**Relational Database (RDB)**
An **RDB Connection** defines a connection to a tier-3 relational database so that your application can update data on the tier-3 database as part of a Component Broker transaction.

**Generic**

A **Generic Connection** defines a generic connection to a tier-3 system other than defined by a HOD Connection, ECI Connection, APPC Connection, or RDB Connection. It is not used in this release, and is intended for future types of connection.

Each application that needs to communicate with a tier-3 system provides the Connection objects it needs. The specific characteristics of the connection are set by editing the Connection object through the System Manager user interface.

The characteristics defined by attributes of a Connection object apply to all connections to the tier-3 system using that Connection object. For example, they limit the maximum number of connections to the tier-3 system for that Connection object.

A connection to a tier-3 system can tie up tier-3 resources until committed or rolled-back at the end of the controlling transaction or session. For example, an ECI Connection represents an extended logical unit of work that acquires exclusive and conversational use of a CICS (non-facility) task. ECI_COMMIT or ECI_ROLLBACK is issued during the session commit or rollback to ensure that the unit of work is properly coordinated with the session.

## Communication Protocols

Component Broker application servers can communicate with tier-3 systems via the following protocols:

- Transmission control protocol/Internet protocol (TCP/IP)
- Advanced program-to-program communication (APPC)

Your application servers can connect to a wide range of tier-3 systems using the TCP/IP protocol; for example, to connect to CICS for OS/2, CICS for Windows NT, and CICS on Open Systems servers.

For ECI connections through a CICS Universal Client for Windows NT, application servers can also use TCP/IP to communicate with CICS on OS/390 through the IBM TCP62 protocol mapper, which allows APPC applications to communicate over a TCP/IP network.

Application servers can communicate with tier-3 systems via APPC, by using IBM Communications Server.

### Related Concepts

“ECI Connections to Tier-3 Systems” on page 31
“HOD Connections to Tier-3 Systems” on page 32
“APPC Connections to Tier-3 Systems” on page 33
“CICS Transaction Gateway” on page 34
“IBM Communications Server” on page 34

### Related Tasks

“Configure a new ECI Connection to a Tier-3 System” on page 348
“Configure a new HOD Connection to a Tier-3 System” on page 344
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354
“Configure the iPAAServices application onto the application server” on page 394

“Configure a Server for Connections to tier-3 Systems” on page 392
“Configure a new Connection to a Database for use by Applications” on page 237

## ECI Connections to Tier-3 Systems

An application program running on a Component Broker application server can use the External Call Interface (ECI) of a CICS Client to call a CICS program located on a tier-3 CICS region; for example, CICS for OS/390 or CICS for Windows NT. This enables the application program to make use of existing CICS routines that could be used, for example, to make enquiries on a database.

Communications between an application server and the tier-3 CICS region are via a “CICS Transaction Gateway” on page 34, as shown in Figure 7.

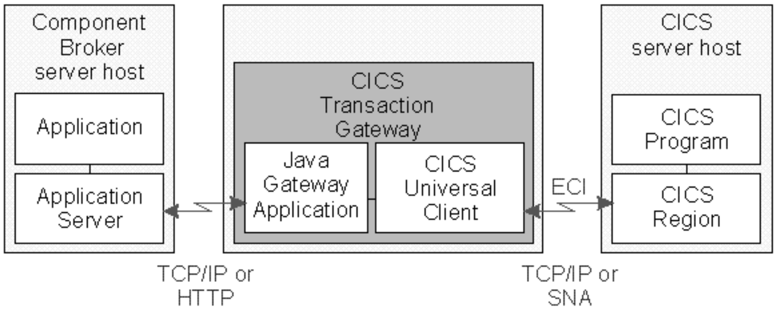


*Figure 7. ECI Connection to a tier-3 CICS Region*

The CICS Transaction Gateway can run on the same workstation as the Component Broker application servers run on or a separate workstation, such as a Web server host. Communication between the application server and the CICS Transaction Gateway on a separate workstation can be either TCP/IP or HTTP.

You can use an existing CICS Transaction Gateway or install and configure a new Gateway for use by Component Broker. The CICS Transaction Gateway is shipped with Component Broker.

The CICS Transaction Gateway can support an unlimited number of concurrent ECI calls to CICS regions, with no restrictions on communication protocols, functions, or whether the calls are to the same or different CICS regions.

An **ECI Connection** is used to configure and manage the parameters of the ECI connection within Component Broker; for example, the name of the CICS region and the address of the CICS Transaction Gateway.

There is no reuse of connections across session boundaries, although the pool limits are managed. When a connection is requested, a new connection is created and allocated.

Use of ECI enables you to develop new client/server applications in which the display and processing logic is appropriately split between the client (application on a Component Broker application server) and the server (application on a CICS region).

**Note:** An ECI Connection represents an extended logical unit of work that acquires exclusive and conversational use of a CICS (non-facility) task. ECI_COMMIT or

ECI_ROLLBACK is issued during the session commit or rollback to ensure that the unit of work is properly coordinated with the session.

### Related Concepts

"HOD Connections to Tier-3 Systems"
"APPC Connections to Tier-3 Systems" on page 33
"Connections to Tier-3 Systems" on page 29

### Related Tasks

"Configure a new ECI Connection to a Tier-3 System" on page 348
"Configure a new HOD Connection to a Tier-3 System" on page 344
"Configure a new APPC Connection to a Tier-3 System" on page 353
"Configure the iPAAServices application onto the application server" on page 394
"Configure a Server for Connections to tier-3 Systems" on page 392

## HOD Connections to Tier-3 Systems

An application program running on a Component Broker application server can use 3270 emulation to call a program located on a tier-3 system; typically, a CICS region or IMS server. It enables you to use 3270 emulation to run, unchanged, existing 3270 applications.

Communications between an application server and the tier-3 system are via Host On-Demand (HOD), as shown in Figure 8.

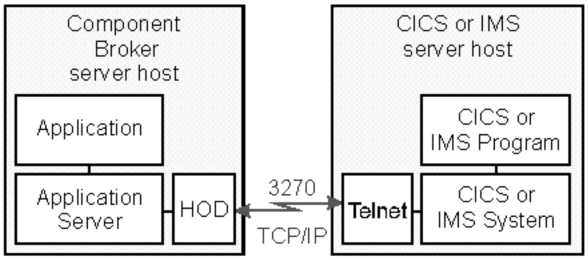


*Figure 8. HOD Connection to a Tier-3 System*

Host On-Demand is a member of the eNetwork software family that is a Java-based solution that incorporates industry-standard Telnet 3270 (TN3270) protocols. Component Broker ships a subset of Host On-Demand, which provides a Java-based TN3270 client, as part of the Procedural Applcation Adaptor (PAA).

HOD connections are allocated, and potentially reused, from a managed pool of resources. The following characteristics of the pool are defined through system management:

- The maximum number of connections to individual tier-3 systems
- The maximum time that competing managed objects are made to wait for the limited number of connections in the pool.

When a new connection is requested at the beginning of a session it is allocated from the pool and logged on with the userid and password predefined for the server. If the pool already contains a dormant connection with the required connection attributes (including the userid), it is allocated to the new request. Otherwise, a new connection is established, and a currently dormant connection may need to be logged off (and even disconnected from the tier-3 system) to allow the new connection to be created in the pool.

### Related Concepts

“ECI Connections to Tier-3 Systems” on page 31
“APPC Connections to Tier-3 Systems”
“Connections to Tier-3 Systems” on page 29

### Related Tasks

“Configure a new HOD Connection to a Tier-3 System” on page 344
“Configure the iPAAServices application onto the application server” on page 394
“Configure a Server for Connections to tier-3 Systems” on page 392

## APPC Connections to Tier-3 Systems

An application program running on a Component Broker application server can use LU6.2 to communicate with tier-3 systems to update data as part of a Component Broker transaction.

APPC communication between an application server and a tier-3 CICS region is via an IBM Communications Server running on the same host as the application server, as shown in Figure 9.

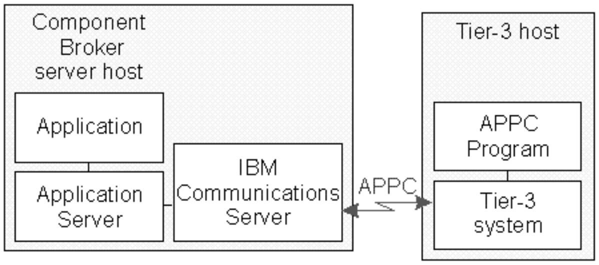


*Figure 9. APPC Connection to a tier-3 CICS Region*

There is no reuse of connections across session boundaries, although the pool limits are managed. When a connection is requested, a new connection is created and allocated.

### Related Concepts

“ECI Connections to Tier-3 Systems” on page 31
“APPC Connections to Tier-3 Systems”
“Connections to Tier-3 Systems” on page 29

### Related Tasks

“Collect information for your SNA configuration” on page 358
“Configure Communications Server” on page 367
“Configure VTAM with details of your Component Broker APPC Connections” on page 388
“Configure the iPAAServices application onto the application server” on page 394

## IBM Communications Server

The IBM Communications Servers for Windows NT and AIX are comprehensive networking products that enable communication between many types of devices across a number of network protocols. Component Broker application servers can use the APPC part of these products to connect to a SNA network. This can be either with or without the APPN support. Throughout the rest of this information IBM Communications Server for Windows NT and IBM Communications Server for AIX are both referred to as *Communications Server*, except where platform-specific information is needed.

- WIN The IBM Communications Server for Windows NT should be installed on each Windows NT host computer where there is a Component Broker application server that is to connect to a SNA network. If your host will be communicating over a Local Area Network (LAN) you must select the IEEE 802.2 IBM LLC2 protocol interface when you install the IBM Communications Server for Windows NT. The *Communications Server for Windows NT Up and Running Guide* explains how to install IBM Communications Server for Windows NT.

  The installation adds a number of new applications to your Windows NT host. The applications that are useful when the IBM Communications Server for Windows NT is being used with Component Broker are:

  **SNA Node Configuration**
  Enables you to create the SNA configuration that describes how application servers connect to the SNA network and to the systems that they communicate with.

  **SNA Node Operations**
  The primary operations facility used to view the status of the SNA network; for example to see which systems the IBM Communications Server for Windows NT is communicating with. It also allows you to connect to, and disconnect from, other systems in the network.

  **Log Viewer**
  Used to view the messages and return codes produced by IBM Communications Server for Windows NT

  **Trace Facility**
  Used for gathering additional information when diagnosing SNA network problems

- AIX The IBM Communications Server for AIX should be installed on each AIX host computer where there is a Component Broker application server that is to connect to a SNA network.

**Related Concepts**

"Introduction to SNA" on page 471

## CICS Transaction Gateway

The IBM CICS Transaction Gateway provides secure, easy access from Web browsers and network computers to business-critical applications running on a CICS Transaction Server or TXSeries server using standard Internet protocols in a range of configurations.

CICS Transaction Gateway is a robust and scalable complement to a Web server, and as such can be implemented as an e-business connector for IBM WebSphere, a runtime environment for Java servlets.

The CICS Transaction Gateway is provided with Component Broker and runs on the OS/2, Windows NT, AIX, and Solaris platforms.

Figure 10 shows how a web-client can access CICS programs and data. Note that the CICS Transaction Gateway is shown as installed on a Web server machine. This is necessary only if you are using the CICS Transaction Gateway with Java applets. An application running on a Component Broker application server on any managed server host can send ECI requests through the CICS Transaction Gateway (like the Java application in the figure).

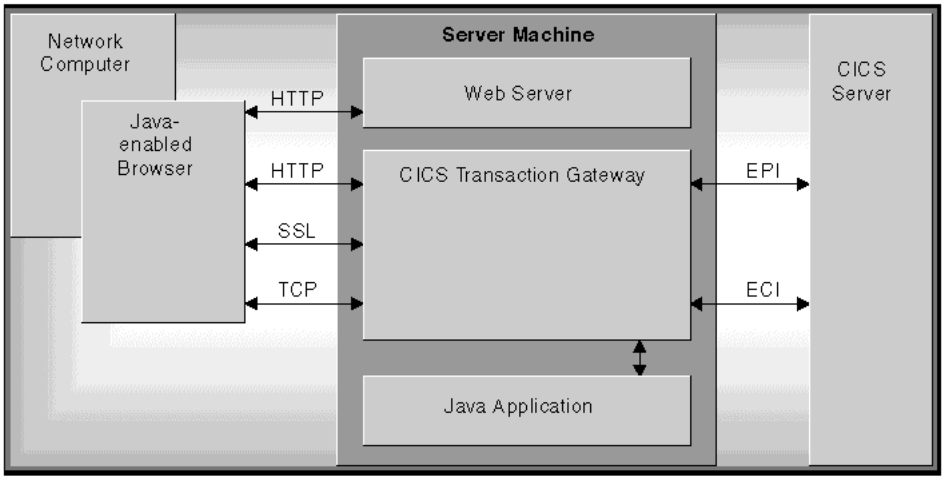


*Figure 10. CICS Transaction Gateway*

The CICS Transaction Gateway provides the following:

1. A **Java gateway application** that is usually resident (for security reasons) on a Web server workstation. It communicates with CICS applications running in CICS servers through interfaces provided by the CICS Universal Clients. This Java application was previously available in the IBM CICS Gateway for Java.
2. A **CICS Universal Client** that provides the ECI and EPI interfaces, as well as terminal emulation function. The **ECI interface** enables a non-CICS Client application to call a CICS program synchronously or asynchronously as a subroutine. The **EPI interface** enables a non-CICS Client application to act as a logical 3270 terminal and so control a CICS 3270 application. The CICS Universal Clients allow communication with CICS Servers over the NetBIOS, TCP/IP, and APPC protocols, depending on the platform.
3. A **CICS Java class library** that includes classes that provide an application programming interface (API), and are used to communicate between the Java gateway application and a Java application (applet or servlet). These Java classes were previously available in the IBM CICS Gateway for Java.
4. A **Terminal Servlet** that allows you to use a Web browser as an emulator for a 3270 CICS application running on any CICS server.
5. A set of **Java EPI Beans** for creating Java front-ends for existing CICS 3270 applications, without any programming.

The CICS Transaction Gateway can concurrently manage many communication links to connected Web browsers, and can control asynchronous conversations to multiple CICS regions. The multithreaded architecture of the CICS Transaction Gateway enables a single Gateway to support multiple concurrently connected users.

The CICS Transaction Gateway monitors a TCP/IP port for incoming requests from Java applications. When a request is received, the gateway parses the data and builds a corresponding ECI or EPI request. Then the gateway invokes its CICS Universal Client through native interfaces. The CICS Universal Client contacts the correct CICS server and executes the request. Any response from the CICS server is received by the CICS Transaction Gateway, which then builds a response and sends it back on the TCP/IP connection which the Java application originally established with the CICS Transaction Gateway.

To configure Component Broker to use the CICS Transaction Gateway, you specify the network address and TCP/IP port of the gateway on each ECI Connection that is to use the gateway, as described in “Configure a new ECI Connection to a Tier-3 System” on page 348.

**Related Concepts**

“ECI Connections to Tier-3 Systems” on page 31

**Related Tasks**

“Configure a Server for Connections to tier-3 Systems” on page 392

## Securing your Enterprise

Information systems, particularly distributed systems, and even moreso those that are exposed to public networks such as the Internet, are inherently at-risk of sabotage, mischief, and subversion. To combat these threats, and to ensure the integrity of the distributed infrastructures on which your mission-critical business applications run, Component Broker supports a variety of security services. These include, primarily, authentication, message protection, and authorization. In the future additional authorization and auditing services will be introduced to further reinforce these objectives.

Component Broker provides the mechanisms and technologies to secure your distributed system. However, your distributed system is no more secure than you make it. This depends on how you set up security and the procedures you employ to administer it. Likewise, if end-users choose weak passwords or leave them lying around they expose the information system.

Securing your information systems can be expensive. Securing these systems usually requires introducing constraints and barriers that necessarily effect the efficiency of your operations. On the other hand, the loss of information or disruptions to your business resulting from malicious or accidental subterfuge can be even more expensive; perhaps even *much* more expensive.

Consequently, you want to employ the right amount of security protection to your systems, striving to form a balance between the threats that face your systems, the risk of any of those threats occurring, the cost to your business if any losses occur, and the impact any security mechanisms you use will have on the efficiency of your operations.

**Related Concepts**

"Security" on page 263
"Security in a Component Broker Network" on page 263

**Related Tasks**

"Chapter 11. Administer Security in your Enterprise" on page 261

## System Management Representation of Your Enterprise

CBConnector System Management is used to define and operate enterprises of *system management objects*. CBConnector System Management groups and represents such objects according to its *object model*, which is also known as the *common data model*.

The *common data model* is a template that describes the structure of "Configuration Data" on page 11 for managed objects within CBConnector System Management. It comprises folders of objects that form a hierarchical tree structure. The model describes:

- The folders that can exist
- The attribute names, types, limits, and default values of objects in those folders
- The relationships that can exist between objects

The common data model can be thought of as comprising the following three conceptual parts, called "worlds", related to the way in which you do system management:

**"The Model World" on page 39**

This defines the topology of an enterprise in high-level terms. For this world, you use *model objects* created within Configurations of your Management Zones to do the following:

- Define the things to be managed (for example, hosts, servers, and applications)
- Define the relationships between managed objects (for example, define which applications are configured for use on servers and which hosts those servers are configured to run on)
- Create logical groups of objects for better management

The model world is the primary interest of system administrators. Model objects are normally created by a system administrator through the CBConnector System Manager user interface. The data is stored in the "The System Manager" on page 10 and can be backed up and restored independently of any other data.

**"The Install World" on page 41**

This defines the topology of applications in terms of the classes and DLLs that applications consist of and the homes and containers that they need to run. *Install objects* are created automatically when you install CBConnector application software, and are used by CBConnector System Management to facilitate system management.

The install world is created by application installation tools. Although the install world can be viewed through the CBConnector System Manager user interface, it cannot be changed by system administrators. Therefore, it is of little interest to system administrators.

The install word is of interest to application developers who need to understand the topology of their applications and to develop the data that creates the Install objects for their applications.

This data is stored in both the "The System Manager" on page 10 and "The SM Agents" on page 11, but is primarily intended for use by SM agents.

**"The Image World" on page 40**

This represents things that exist in the real world. For example, a *Server Image* represents a server that exists on a host computer.

The image world is created automatically by CBConnector System Management from the model and install worlds. It is of some interest to system administrators, mainly for monitoring running servers, clients, applications, and other active parts of the enterprise. The topology of the image world cannot be changed except by updating it from the model world. However, system administrators can take some actions directly on Image objects and can change some attributes of Image objects. For example, a system administrator can stop and start servers, but to create a new server would have to use the model world.

The images of objects on a host exist in the "The SM Agents" on page 11 on that host.

The System Manager and SM agents use the common data model to organize their configuration data.

Some objects, such as Containers, are managed as objects in both the Model and Image worlds. Such objects are matched by name and allow an application package to supply the relationships of the corresponding Image object, while the System Administrator supplies the attribute values through the associated definitional object.

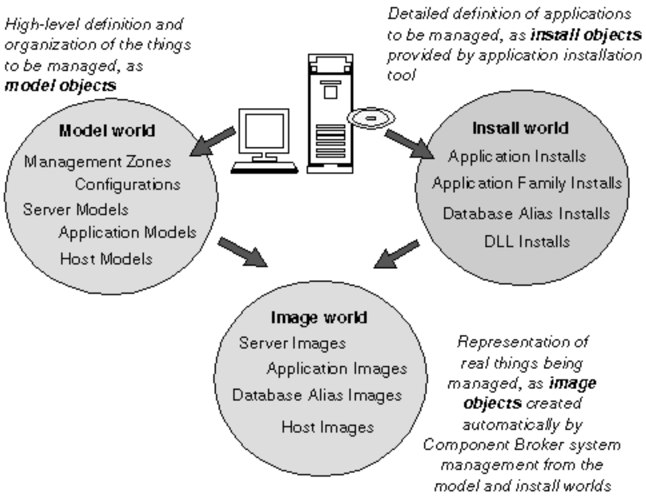

High-level definition and organization of the things to be managed, as **model objects**
Detailed definition of applications to be managed, as **install objects** provided by application installation tool
Model world
Management Zones
Configurations
Server Models
Application Models
Host Models
Install world
Application Installs
Application Family Installs
Database Alias Installs
DLL Installs
Image world
Server Images
Application Images
Database Alias Images
Host Images
Representation of real things being managed, as **image objects** created automatically by Component Broker system management from the model and install worlds

*Figure 11. The Model, Image, and Install worlds*

**Related Concepts**

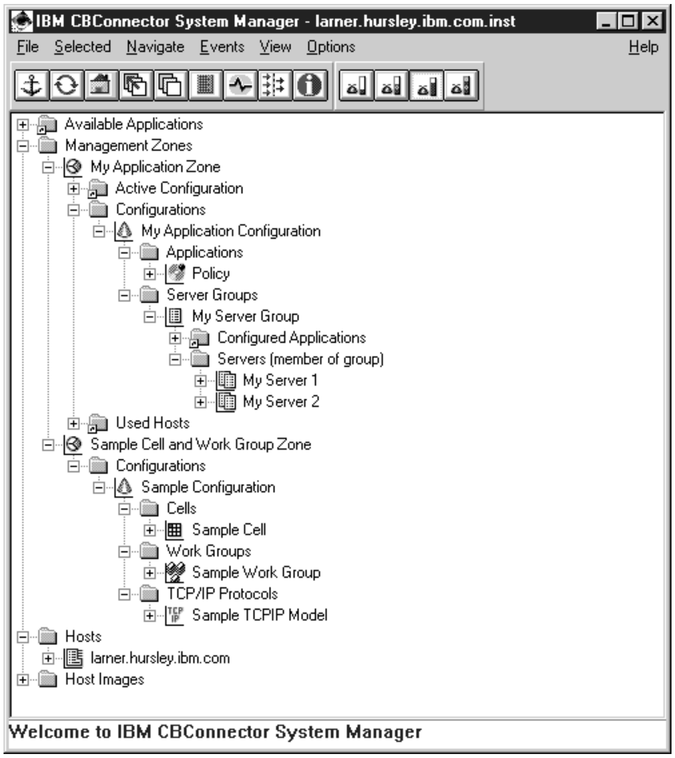
“The Model World”
“The Image World” on page 40
“The Install World” on page 41
“System Management Objects used to Define your Enterprise” on page 42
“System Management Objects used to Operate your Enterprise” on page 45
“The System Manager” on page 10
“The SM Agents” on page 11

## The Model World

You can use CBConnector System Management to administer your enterprise according to your business needs rather than according to the hosts on which servers, clients, and applications exist. To do this, you define the things that you want to administer as *model objects* in one or more *management zones*. (See Figure 12.) (These model objects form the "model world" of your enterprise.)

The model world is the primary interest of system administrators. Model objects are normally created by a system administrator through the CBConnector System Management user interface. The data is stored in the *“The System Manager” on page 10* and can be backed up and restored independently of any other data.

The term *model* is used to refer generically to all the classes of objects used define an enterprise.



Welcome to IBM CBConnector System Manager

*Figure 12. CBConnector System Management Model Objects*

**Related Concepts**

“The Image World” on page 40

“The Install World” on page 41
“System Management Objects used to Define your Enterprise” on page 42
“System Management Objects used to Operate your Enterprise” on page 45
“Management Zones” on page 17
“Configurations” on page 18
“Cells and Work Groups” on page 19
“Hosts” on page 20
“Application Servers and Server Groups” on page 25
“Client Styles” on page 26
“Applications” on page 27
“Database Aliases” on page 28

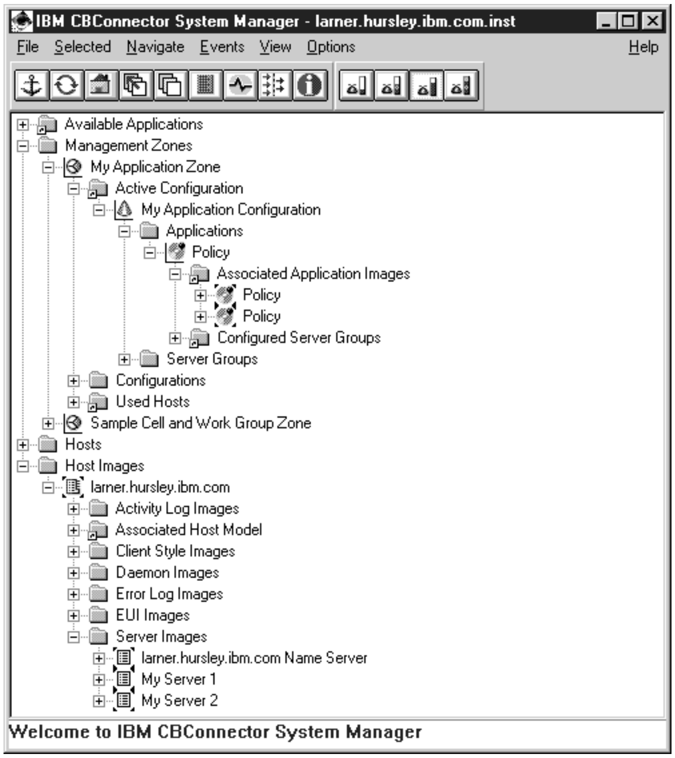
## The Image World

*Images* are the representations of objects, such as servers and applications, that exist in the real world. For example, a *Server Image* represents a server that exists on a host computer.

The image world is created automatically by CBConnector System Management from the model and install worlds. It is of some interest to system administrators, mainly for monitoring running servers, clients, applications, and other active parts of the enterprise. The topology of the image world cannot be changed except by updating it from the model world. However, system administrators can take some actions directly on Image objects and can change some attributes of Image objects. For example, a system administrator can stop and start servers, but to create a new server would have to use the model world.

The images of objects on a host exist in the “The SM Agents” on page 11 on that host.

Images for objects are grouped under the Host Image that represents the host on which the objects exist. For example, Figure 13 on page 41 shows a Host Image, and the Images for servers on that host. Images can also be accessed through relationships from their associated model objects.



Welcome to IBM CBConnector System Manager

*Figure 13. CBConnector System Management Image objects*

**Related Concepts**

"The SM Agents" on page 11
"The Model World" on page 39
"The Install World"
"Application Servers and Server Groups" on page 25
"Applications" on page 27

## The Install World

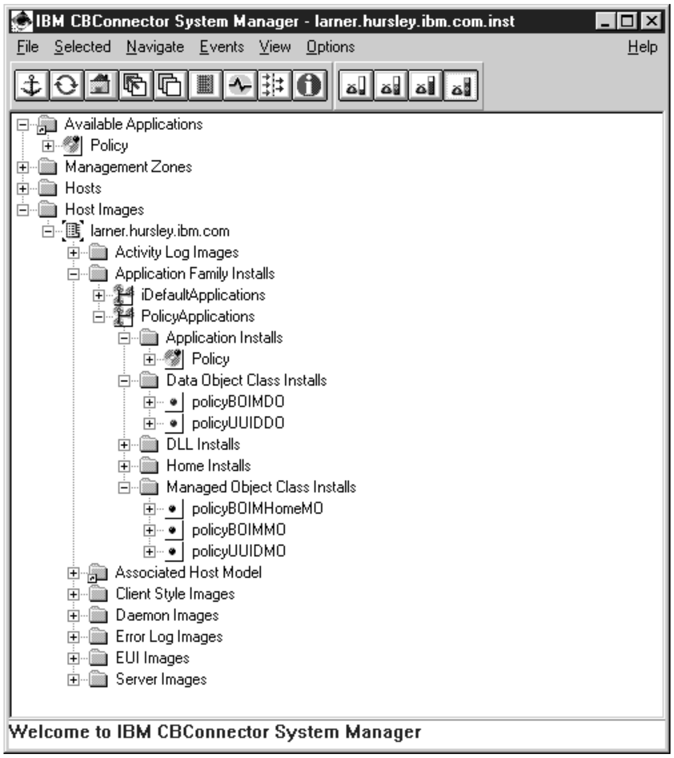
*Install objects* represent the software elements installed on a host for CBConnector applications. This data is stored in both the "The System Manager" on page 10 and the "The SM Agents" on page 11, but is mainly used by the agents. These objects are created automatically when you install CBConnector application software, and are used by CBConnector System Management to enable system management of applications. For example, the install objects for an application do the following:

- Provide appropriate attributes for creating an Application Image and other related Image objects
- Define classes, DLLs, homes, and containers used by the application

The install world is created by application installation tools. Although the install world can be viewed through the CBConnector System Management user interface, it cannot be changed by system administrators. Therefore, it is of little interest to system administrators.

The install word is of interest to application developers who need to understand the topology of their applications and to develop the data that creates the Install objects for their applications.

Install objects are grouped under the Host Image that represents the host on which the software is installed. For example, Figure 14 shows a Host Image and some of the Install objects for application software installed on that host.



Welcome to IBM CBConnector System Manager

*Figure 14. CBConnector System Management Install objects*

### Related Concepts

“The SM Agents” on page 11
“Installing Applications” on page 222
“The Model World” on page 39
“The Image World” on page 40
“Application Servers and Server Groups” on page 25
“Applications” on page 27

### Related Tasks

“Load a new Application” on page 141

## System Management Objects used to Define your Enterprise

The main types of objects that you can use to define your enterprise to CBConnector System Management are contained within the Management Zones folder of the Home view, as shown in Figure 15 on page 43. These objects, which form the CBConnector System Management *“The Model World” on page 39*, are outlined after the figure.

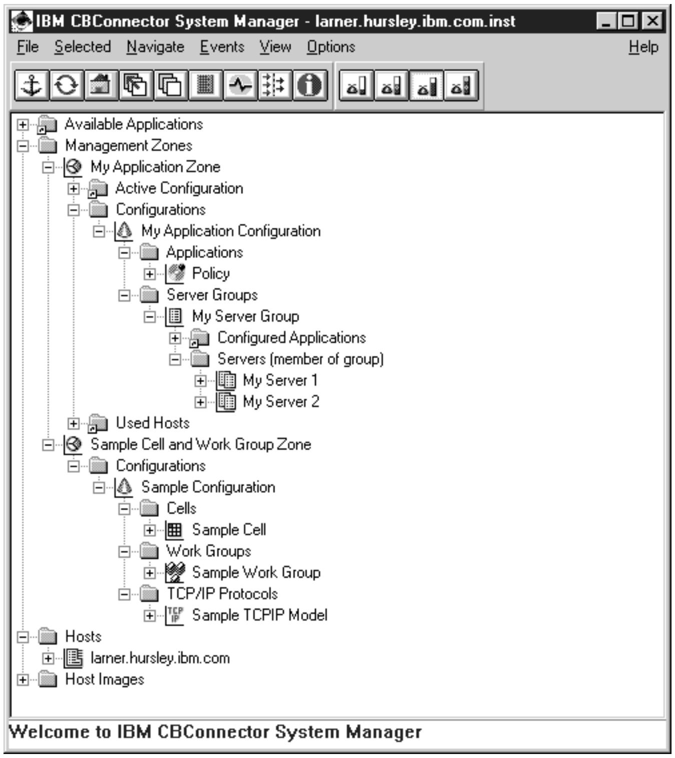


*Figure 15. Some objects of the CBConnector System Management Model World*

The main objects in the model world are outlined below. Details about these, and other model object classes, are provided in the Object Reference.

**Note:** If any of these folders are not visible, the folder may be empty or you may not have an appropriate user-level. To change what folders are displayed, see “Control Which Objects are Displayed” on page 62.

**Management Zones**

This folder can contain one or more Management Zones that you use to group object definitions to be administered as a unit. You must define a Management zone before you can define (within it) configurations.

You can access this folder directly from the **Home** view of an Information Controller window.

**Configurations**

This folder can contain one or more Configurations that you use to create alternative topologies within a Management zone. You must define a Configuration before you can define (within it) any Model objects; for example, Server Groups.

You can access this folder by expanding or opening a Management zone.

**Server Groups**

This folder can contain one or more models that you use to define the attributes of groups of servers. You use a Server Group to:

- Define the attributes common to one or more servers, which are represented by *Servers (member of group)* within the Server Group
- Collect the Applications to be used by the servers in the group
- Define workload management characteristics for a controlled server group

- Create special servers for a controlled server group

You can access this folder by expanding or opening a Configuration.

**Servers (member of group)**
This folder can contain definitions for one or more application servers that are members of the same server group. The servers typically have the same characteristics and always support the same applications. Workload balancing can also be used to distribute the workload across the members of the server group.

You can access this folder by expanding or opening the Server Group.

**Applications**
This folder can contain one or more models that you use to represent business applications. You use an Application to identify the business applications to be used on servers defined by Server Groups and Servers (free standing).

You can access this folder by expanding or opening a Configuration.

**Database Aliases**
This folder can contain one or more models that you use to define the databases to be used by applications. You can access this folder by expanding or opening a Configuration.

**Client Styles**
This folder can contain one or more models that you use to define the attributes of clients. You use a Client Style to:

- Define the attributes common to one or more clients
- Group the Client Applications to be used by the clients.

You can access this folder by expanding or opening a Configuration.

**Client Applications**
This folder can contain one or more models that you use to represent business applications to run on clients defined by Client Styles.

You can access this folder by expanding or opening a Configuration.

**Hosts** This folder contains an object for each host that you specified when installing CBConnector System Management.

You can access this folder directly from the Home view.

**Protocols**
This folder can contain one or more models that define the communication protocols provided by hosts.

You can access this folder by expanding or opening a Configuration.

**Cells** Use this folder to identify the cells that your hosts belong to. For example, you can configure new Hosts into the Component Broker cell.

You can access this folder by expanding or opening a Configuration of the Management Zone that define syour host environment.

**Work Groups**
Use this folder to identify the workgroups that your hosts belong to. For example, you can define new workgroups and can configure new Hosts into them.

You can access this folder by expanding or opening a Configuration.

**Note:** The classes of objects are named with initial capitals without the *model* term; for example, **Server Group**. This naming convention is used by the CBConnector System Management user interface and throughout the associated information.

In the descriptions of administration tasks, it is assumed that you know how to navigate to (display) an object to be acted on. The path from the Home view to each class of object is given in the description of the object class in the Object Reference. Generally, for administrative tasks, you use the model objects that can be found by expanding the Management Zones folder of the Home view, and if required expanding folders that it contains.

If a folder is not displayed when you expect to see it, the folder may hidden by the filter or your user-level. Reset the filter or your user-level to enable the folder to be displayed. If the View panel displays several folders and objects, you may have to scroll the panel up or down until the object that you want to act on is displayed. Consider using two Information Controller windows to reduce the amount of information displayed in any one window. (See Display a New Information Controller Window.)

In particular, you may find it useful to have Information Controller windows displaying the following types of information:

- A window displaying the Server Groups folder as its root. This enables you to see easily the definition of each server group in terms of the applications configured on a group and the servers that are members of it.
- Separate Information Controller windows displaying, in Text View, folders that contain large numbers of objects. This is better than expanding those folders in the Home tree view, because you can keep the windows much more compact and thus more easily used. For example, you might have several windows that each display, in Text View, the trees for a separate Host.

**Related Concepts**

“The Model World” on page 39
“Appendix A. Features of the User Interface” on page 433
“Generic Actions” on page 69

**Related Tasks**

“Display Objects” on page 64
“Act on Objects” on page 66
“Edit Objects” on page 72
“Control Which Objects are Displayed” on page 62

## System Management Objects used to Operate your Enterprise

When you activate a Configuration of one of your Management Zones, the System Manager creates or updates the runtime configuration of that Management Zone in your enterprise.

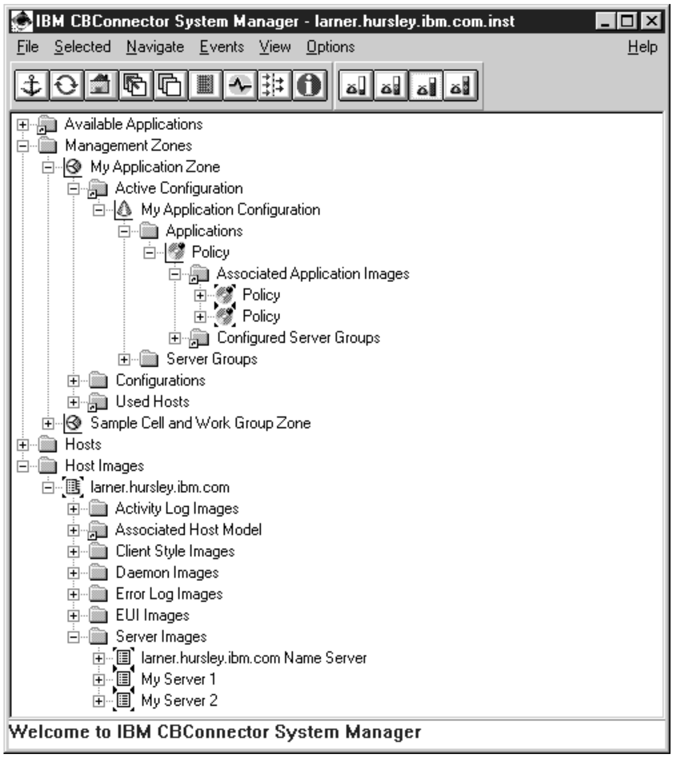
The runtime configuration is represented by the *Active Configuration* within the Management Zone and by *Image* objects within the Host Images for hosts used by the Management Zone. The Images for things on a host exist within the SM Agent on that host.

The Active Configuration provides a view of the runtime configuration across all hosts used by the Management Zone. The Host Images provide a view of the runtime configuration on the one host.

The System Manager user interface displays folders of Images within the related Host Image, and in *Associated xxx Images* folders within the Active Configurations of Management Zones, Each of these folders (for example, *Server Images*) can contain one or more images of a specific class.

You can monitor and act on the runtime configuration of your enterprise, by using the system management Image objects.

You can access Images through shortcut icons from related model objects or by expanding or opening the Host Images folder, as shown in Figure 16.



Welcome to IBM CBConnector System Manager

*Figure 16. CBConnector System Management images*

In the descriptions of operation procedures, it is assumed that you know how to navigate to (display) an object to be acted on. The path from the Home view to each class of object is given in the Object Reference.

Generally, for operation procedures, you act on Images by their shortcut icons from related model objects; for example, to stop the Policy application you can use the **Stop** action on the Policy shortcut icon within the *Associated Application Images* folder of the Application model. (See Figure 16.)

If you want to act on an Image object directly, you can do so by expanding the Host Images folder of the Home view, and if required expanding folders that it contains. To do this, you must have an appropriate “Control Which Objects are Displayed” on page 62.

If a folder is not displayed when you expect to see it, you may need to select a different user-level setting or reset the filter. (See “Control Which Objects are Displayed” on page 62.) If the View panel displays a lot of folders and objects, you may have to scroll the panel up or down until the object that you want to act on is displayed. Consider using two Information Controller windows.

**Related Concepts**

“The Image World” on page 40
“System Management Representation of Your Enterprise” on page 37
“Example: Operate a Server” on page 201
“Operating CBConnector System Management Components” on page 431

**Related Tasks**

“Monitor Events” on page 419
“Display Runtime Information about your Enterprise” on page 422
“Display Information in Component Broker Logs” on page 423
“Control Component Broker Trace” on page 426
“Change the Active Configuration of Your Enterprise” on page 431

## Integrate Component Broker System Management with Tivoli

To enable Component Broker system management integration with Tivoli, you complete the following tasks:

1. “Install and Configure the Component Broker plus module for Tivoli” to add icons to Tivoli, enable its event server to recognize Component Broker events, and define the Tivoli event consoles that are to monitor those events.
2. “configure and enable Tivoli event monitoring” on page 50 so that Component Broker sends events to the Tivoli event server whenever an entry is added to a Component Broker error log.
3. (Optional) If you want Tivoli to be able to launch the Component Broker System Manager user interface, you need to “Install a System Manager User Interface” on page 51 on the Tivoli host.

**Related Concepts**

“Integrated System Management with Tivoli” on page 8

**Related Tasks**

“Display Component Broker Events Using the Tivoli Event Manager” on page 424

## Install and Configure the Component Broker plus module for Tivoli

Use this procedure to install the Component Broker plus module for Tivoli and configure it to enable the Tivoli event server to recognize Component Broker events and define the Tivoli event consoles that are to monitor those events.

**To install and configure the Component Broker plus module for Tivoli, complete the following steps:**

1. Ensure that you are logged in as an administrator with the ″super″ role.

   The initial install of any Tivoli plus module needs to be done by an administrator with the ″super″ role. If you install without this role, the plus module will not show up on the Tivoli desktop. If you install the plus module and do not see the Plus icon, login as an administrator with ″super″, and run $LINKDIR/wcrttopcol (which creates the TivoliPlus collection on a desktop). You should not have to cycle the oserv, or reboot the machine; the icon should immediately show up on the Tivoli desktop.

2. Install the Component Broker plus module for Tivoli, from the Component Broker supplemental compact disk.

   You do this using the standard install process for Tivoli plus modules. For example;

   a. From the TME 10 desktop, click **Desktop - Install - Install Product**
   b. On the Install Product dialog window, click the **Select Media** button
   c. From the File Browser dialog window, select the \Tivoli directory on the supplemental compact disk
   d. Close the File Browser dialog window. The Install Product dialog window should show ″Component Broker Plus for Tivoli ...″ in the top window
   e. Select the Tivoli managed nodes where the Component Broker System Manager user interface may be activated and move them into the ″Clients to install on″ list. The Component Broker plus module for Tivoli *does not* have to be installed onto the Component Broker hosts. The Component Broker code already has the code to communicate with Tivoli.

      **Note:** You need to install the Component Broker System Manager user interface on the Tivoli hosts that may want to activate that user interface. You do this by running the Component Broker setup application and selecting ″Custom Install″ and ″System Manager User Interface″. There is no need to install any of the Component Broker prerequisite products (for example, DCE). *The System Manager user interface is the only Component Broker component supported on the same host as Tivoli.*
   f. On the Install Product dialog window, click the **Install & Close** button.

   When the install process has finished, there should be a Tivoli Plus icon on the TME 10 Desktop. You can then go on to configure the Tivoli Plus module for Component Broker, by completing the following steps:

3. Double-click on the Tivoli Plus icon to open the TivoliPlus window. In that window, you should see the ″Component Broker Plus for Tivoli″ icon.
4. Double-click on the ″Component Broker Plus for Tivoli″ icon to open the Component Broker Plus for Tivoli window with icons for the various Component Broker actions. For example, there should be an icon for starting the System Manager user interface.
5. Double-click the Setup TEC Event Server icon to display the configuration dialog window, as shown in Figure 17 on page 50. You can use this window to configure the Tivoli event server to recognize Component Broker events and to define the event consoles that are to monitor those events.
6. On the configuration dialog window, set the following parameters:

   **use_new_or_old_rulebase**
   Specify **new** to create a new Tivoli rulebase for Component Broker events or **old** if Component Broker events should be added to an existing rulebase.

   **rulebase_name**
   If you set **use_new_or_old_rulebase** to **new**, specify a new Tivoli rulebase name. If you set **use_new_or_old_rulebase** to **old**, specify the name of an existing rulebase.

**clone_rulebase_name**
If you set **use_new_or_old_rulebase** to **new**, this identifies an existing Tivoli rulebase that will be copied to form the new rulebase for Component Broker events. If you set **use_new_or_old_rulebase** to **old**, this field is ignored.

**event_console_name**
(Optional) The name of an existing Tivoli event onsole that should receive Component Broker events.

**rulebase_path**
If you set **use_new_or_old_rulebase** to **new**, specify the file system path where the new Tivoli rulebase should be created. The default path name is *\Tivoli\bin\generic_unix\Tme\Plus\COMPONENT_BROKER\<rulebase_path>*

**Note:** Tivoli requires that if you specify any of the above parameters, then you must specify values for all preceding parameters.

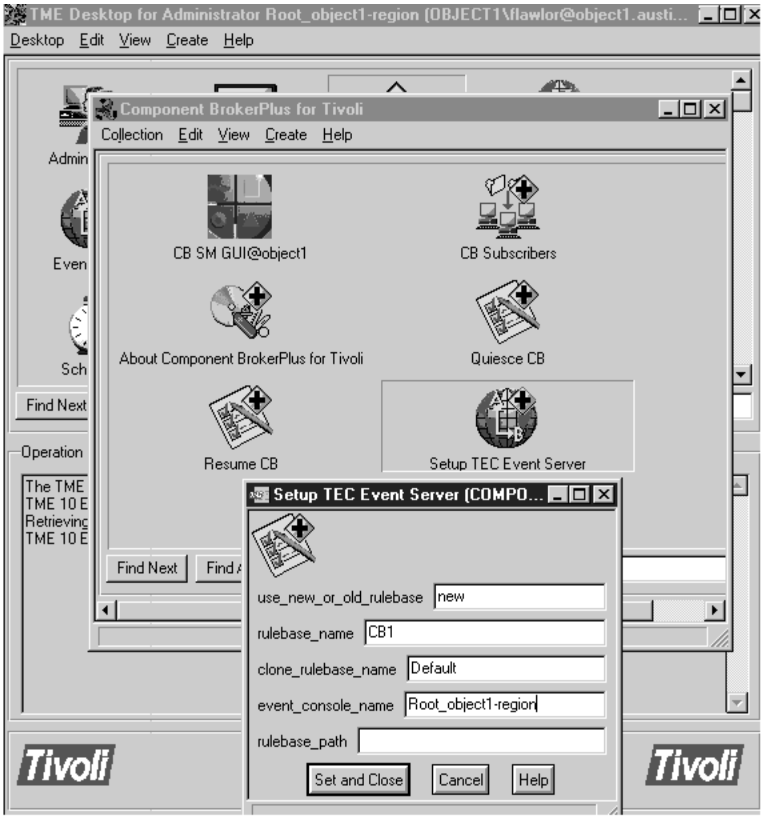
7. The Quiesce and Resume Tivoli tasks are run on the hosts defined in the **Tivoli CB Plus Host** profile icon. This should default on install to the host where the plus module is installed. If it should be a different host, modify the subscribers list as needed.

   **Note:** A Tivoli bug on some Windows NT systems causes the task to also run on the local host. Since this is probably the same as the subscribing host, it effectively is run twice. In this case, you can remove the subscriber in the Tivoli CB Host list to make it run only once.

   **IMPORTANT NOTES:**

   - Tivoli tasks log in and run as a user on the subscribing machines. This is typically set up by Tivoli installation as root on unix machines and a user with administrative priveledges on Windows NT systems. The tasks use the setting of the SOMCBASE environment variable to find CBConnector.
     - WIN SOMCBASE must be set for the Windows NT Tivoli user.
     - AIX SOMCBASE defaults to /usr/lpp/CBConnector if not set.
   - AIX Setting the DISPLAY variable controls where the Component Broker System Manager user interface is displayed. If DISPLAY is not set, the launch script sets it to ″:0″ (the local host).

To enable Component Broker to send events to the Tivoli event server whenever an entry is added to the Component Broker error log, “configure and enable Tivoli event monitoring” on page 50.


TME Desktop for Administrator Root_object1-region (OBJECT1\flawlor@object1.austi...
Desktop Edit View Create Help
Admin
Even
Sch
Find Next
Operation
The TME
TME 10 E
Retrieving
TME 10 E
Component BrokerPlus for Tivoli
Collection Edit View Create Help
CB SM GUI@object1
CB Subscribers
About Component BrokerPlus for Tivoli
Quiesce CB
Resume CB
Setup TEC Event Server
Find Next
Find A
Setup TEC Event Server (COMPO...
Set and Close
Cancel
Help
Tivoli
Tivoli

use_new_or_old_rulebase: new

rulebase_name: CB1

clone_rulebase_name: Default

event_console_name: Root_object1-region

rulebase_path:

*Figure 17. The Tivoli Setup TEC Event Server window.*
*This shows the Tivoli Setup TEC Event Server window used for Component Broker.*

**Related Concepts**

“Integrated System Management with Tivoli” on page 8

**Related Tasks**

“configure and enable Tivoli event monitoring”
“Display Component Broker Events Using the Tivoli Event Manager” on page 424
“Install a System Manager User Interface” on page 51 (on a Tivoli host)

## configure and enable Tivoli event monitoring

Use this procedure to enable Tivoli to monitor events for a Component Broker managed host. If you complete this procedure, Component Broker will send a Tivoli event to the Tivoli event server whenever an entry is added to the Component Broker error log on the managed host.

You should complete this task after you have “Install and Configure the Component Broker plus module for Tivoli” on page 47.

You should complete this procedure for each managed host that can issue Component Broker events that you want Tivoli to monitor. Typically this will be hosts on which you are running Component Broker application servers.

**To configure and turn on Component Broker event generation for Tivoli, use the System Manager user interface to complete the following steps:**

1. Display the Host Image for the managed host
2. From the pop-up menu of the Host Image, click **Edit**, to display the Object Editor window
3. Set the following attributes:

   **enable Tivoli event monitoring**
   Set this to **Yes** to instruct Component Broker to send Tivoli events to the Tivoli event server whenever an entry is added to the Component Broker error log.

   **Tivoli event server hostname**
   Specify the hostname of the Tivoli event server host to which Component Broker events should be sent

   **Tivoli event server port number**
   Specify the port number of the Tivoli event server host to which Component Broker events should be sent, as a value other than 0 (zero). You cannot specify port 0 to allow the Tivoli portmapper function to pick a port. The default port number depends on the host platform, as follows: 5529 ( WIN ) or 0 ( AIX )
4. To apply the changes and close the Object Editor window, click **OK**

If you have enabled Tivoli event monitoring for Component Broker and installed and configured it correctly, Component Broker will send Tivoli events to the Tivoli event server whenever an entry is added to the Component Broker error log on the managed host.

**Related Concepts**

"Integrated System Management with Tivoli" on page 8

**Related Tasks**

"Install and Configure the Component Broker plus module for Tivoli" on page 47
"Display Component Broker Events Using the Tivoli Event Manager" on page 424

## Install a System Manager User Interface

WIN Use this procedure to install the System Manager User Interface (only) on a Windows NT machine.

**For the most up-to-date information about installing Component Broker, see the** *Quick Beginnings* and *Planning, Performance, and Installation Guide*.

You only need to complete this procedure if you want to install the System Manager user interface *only* on a host computer; for example, to add that user interface to a Tivoli host.

**Prerequisites:**

- To enable Tivoli to launch the Component Broker System Manager user interface, you must install that user interface on the Tivoli host, as described in this procedure.
- *Do not install other Component Broker components on a Tivoli host*.
- The Component Broker System Manager user interface has been found to cause problems with older versions of Sybase (for example, Version 11). This is due to

Sybase incorrectly handling the LANG environment variable. You can overcome this problem by deleting the LANG environment variable. If you are using a non-EN_US language, you may need to set the LOCPATH and NLSPATH environment variables to compensate for the lack of the LANG environment variable.

- Before you use the CBConnector installation tool, check that the following prerequisites are satisfied:
  - You must already have installed Windows NT onto the host
  - You need the CBConnector installation compact disk
  - You need administrator authority for the host onto which you are installing the CBConnector System Management components

**To install the System Manager user interface host, complete the following steps:**

1. Insert the CBConnector compact disc into your CD-ROM drive. This should start the CBConnector installation tool automatically, and displays the **Welcome** panel.

   Otherwise, you can 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.
2. 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.
3. In the Choose Destination Location window, either accept the default location (`C:\CBroker`) or specify a new location into which Component Broker files will be installed.

   To change the drive or directory, click the **Browse** button to open the Choose Directory window, then in that window completet the following steps:
   a. Type a directory name.
   b. Click the **OK** button to continue.
   c. If the specified directory does not exist, Setup displays a window asking if you want the directory to be created. Click the **Yes** button.
   d. The Choose Destination Location window is displayed again and lists the directory you specified.

   Click the **Next** button to continue.
4. In the Select Installation Type window:
   a. Select the Custom Install installation option.
   b. Click the **Next** button to continue.
5. In the Select the package(s) for installation window:
   a. Select the **System Manager user interface** and **Documentation** check boxes.
   b. Click the **Next** button to continue.
6. In the Enter NetQuestion Install Path window, either accept the default location (`C:\IMNNQ`) or specify a new location into which NetQuestion files will be installed.

   To change the drive or directory, specify a different directory in the **Install Path** field.

7. 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.
8. 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 installation.
9. A status bar is displayed indicating the progress of each component file transfer. The installation may take several minutes.
10. 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.
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.

When your computer restarts, you need to configure Component Broker with details of the System Manager that the user interface normally connects to. Before beginning the configuration, verify the following:

- The JDK classes.zip is included in the CLASSPATH environment variable
- The JDK `bin` directory is in the PATH environment variable

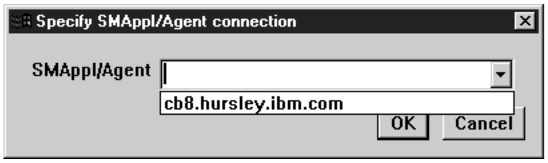
To complete the initial configuration of Component Broker, complete the following steps:

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 **Yes** radio button to configure your entire Component Broker installation.
   b. Select the **Next** button to continue.
3. **If you are configuring specific packages**

   In the Custom Configuration window:
   a. Select one or more packages to configure. You can only select packages that are installed.
   b. Click the **Next** button to continue.
4. 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 used for system management communication. The System Manager user interface will use this port number to communicate with the System Manager. The default port ID is 20002. You can accept the default or supply a different port ID. The number must match the port number that you specified when you installed the System Manager.
   c. Click the **Next** button to continue.
5. 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.
6. 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.
7. An informational window is displayed stating that the system management configuration completed successfully. Click the **OK** button to continue.
8. An informational window is displayed stating that the configuration has completed. Click the **OK** button to close the window.
9. Exit the Component Broker Configuration tool. In the Configuration Status window, click either the **Close** button or the **Cancel** button.

**Note:** If the System Manager user interface cannot connect to the System Manager automatically, it will prompt you for the full hostname of the System Manager (or SM Agent) that you want to connect the user interface to.



If you want Tivoli to be able to start the Component Broker System Manager user interface, you must “Install and Configure the Component Broker plus module for Tivoli” on page 47.

### Related Concepts

“Components used for System Management” on page 9

### Related Tasks

“Install and Configure the Component Broker plus module for Tivoli” on page 47
“configure and enable Tivoli event monitoring” on page 50

# Chapter 2. Use the System Manager User Interface

This topic describes how to start the user interface and to use it to find and act on objects displayed. It contains the following topics:

- “Start or Stop the System Manager User Interface”
- “Change the Appearance of the SM User Interface” on page 60
- “Control Which Objects are Displayed” on page 62
- “Display Objects” on page 64
- “Create Objects” on page 65
- “Select and Deselect Objects” on page 65
- “Act on Objects” on page 66
- “Edit Objects” on page 72
- “Display Help Information” on page 73
- “Example: Explore the User Interface” on page 74

It also provides reference information about “Drag and Drop Actions” on page 78.

For more information about the user interface, see “Appendix A. Features of the User Interface” on page 433. Also, context-sensitive information is available with the user interface.

For information about completing specific tasks using the user interface, see the related tasks below.

**Related Concepts**

“Appendix A. Features of the User Interface” on page 433

**Related Tasks**

“Chapter 3. Create a New Single-Host Enterprise” on page 85
“Chapter 4. Create a New Multi-Host Enterprise” on page 101
“Chapter 5. Configure a new Application Environment” on page 133
“Chapter 6. Administer your Host Environment” on page 167
“Chapter 7. Administer Application Servers” on page 181
“Chapter 8. Administer Clients” on page 205
“Chapter 9. Administer Applications” on page 221
“Chapter 11. Administer Security in your Enterprise” on page 261
“Chapter 12. Administer Workload Management” on page 333
“Chapter 13. Administer Connections to Tier-3 Systems” on page 343
“Chapter 14. Administer Component Broker Services” on page 397
“Chapter 15. Operate your Enterprise” on page 413

## Start or Stop the System Manager User Interface

The procedure that you use to start the System Manager user interface depends on the operating system that you are using, as follows:

- WNT “Start the System Manager User Interface on Windows NT” on page 56.

- AIX “Start the System Manager User Interface on AIX” on page 58.

To exit the user interface, you close the last Information Controller window as described in “Exit the User Interface” on page 60. This closes all the windows of the System Manager user interface.

**Related Concepts**

“System Manager User Interface” on page 12
“Appendix A. Features of the User Interface” on page 433
“Chapter 2. Use the System Manager User Interface” on page 55

**Related Tasks**

“Change the Appearance of the SM User Interface” on page 60
“Install a System Manager User Interface” on page 51
“Install and Configure the Component Broker plus module for Tivoli” on page 47

## Start the System Manager User Interface on Windows NT

WIN Use this procedure to start the Component Broker System Manager user interface to run on Windows NT.

**Notes:**

- You need a personal computer running Windows NT, connected using TCP/IP to the host on which the System Manager runs.
- You must have already installed the System Manager user interface onto the AIX host where you want to start the user interface. This is normally achieved by the typical or system management options of Component Broker installation. Otherwise, you can install the System Manager user interface on its own. For more information, see Installing Component Broker System Management Components.

**To start the user interface, complete the following steps:**

1. Click **Start** on the taskbar
2. Point to **Programs**
3. Point to **IBM Component Broker for Windows NT**
4. Click **System Manager User Interface**

When you start the System Manager user interface, the following are displayed:

1. The Component Broker copyright message window, briefly.
2. The Information Controller window, as shown in Figure 18 on page 57, displays the **Home** view of the system management object network. This window is the main window for user interaction with the System Manager.

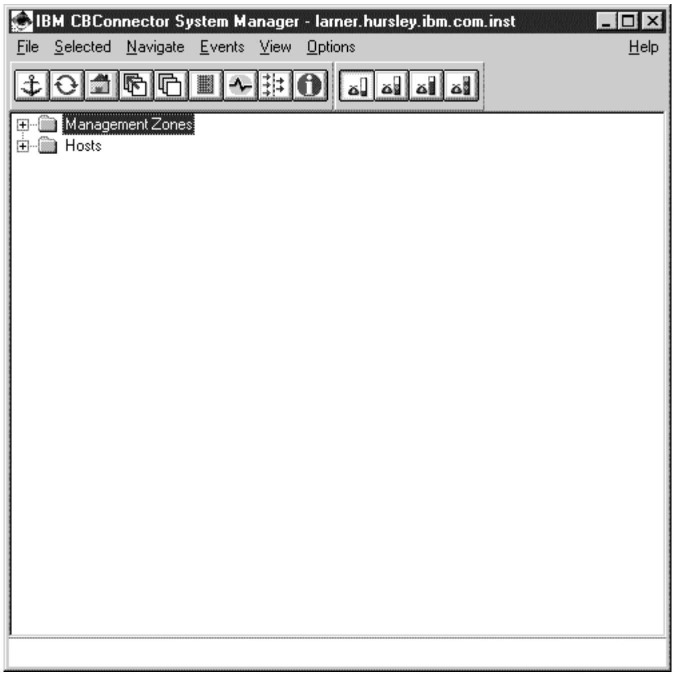
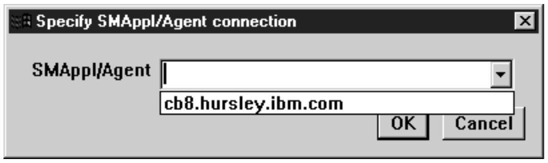


*Figure 18. The Information Controller window. Showing the Home view of the system management object network.*

**Notes:**

1. If the System Manager user interface cannot connect to the System Manager automatically, it will prompt you for the full hostname of the System Manager (or SM Agent) that you want to connect the user interface to.



SMAppl/Agent cb8.hursley.ibm.com

2. If you want Tivoli to be able to start the Component Broker System Manager user interface, you must “Install and Configure the Component Broker plus module for Tivoli” on page 47 and “Install a System Manager User Interface” on page 51 on the Tivoli host.

### Related Concepts

“System Manager User Interface” on page 12
“Appendix A. Features of the User Interface” on page 433
“Operating CBConnector System Management Components” on page 431

### Related Tasks

“Change the Appearance of the SM User Interface” on page 60
“Start the System Manager User Interface on AIX” on page 58
“Install a System Manager User Interface” on page 51
“Install and Configure the Component Broker plus module for Tivoli” on page 47
“Chapter 2. Use the System Manager User Interface” on page 55

## Start the System Manager User Interface on AIX

AIX Use this procedure to start the Component Broker System Manager user interface to run on AIX.

**Prerequisites:**

- You need an Xstation connected using TCP/IP to the host on which the System Manager runs, with a color display and a mouse. You should also ensure that appropriate fonts and colors have been specified and are available. You can specify your own fonts and colors in your .Xdefaults file. If unsure, start the user interface and check the fonts and colors displayed. If they are not appropriate, “Specify Colors and Fonts for the System Manager User Interface on AIX” on page 61 then restart the user interface.

  **Note:** This is particularly important if you are using DBCS characters.
- You must have already installed the System Manager user interface onto the AIX host where you want to start the user interface. This is normally achieved by the typical or system management options of Component Broker installation. Otherwise, you can install the System Manager user interface on its own. For more information, see Installing Component Broker System Management Components.

**To start the user interface**, click the icon for the System Manager User Interface on your desktop, or type the following command at an AIX prompt:

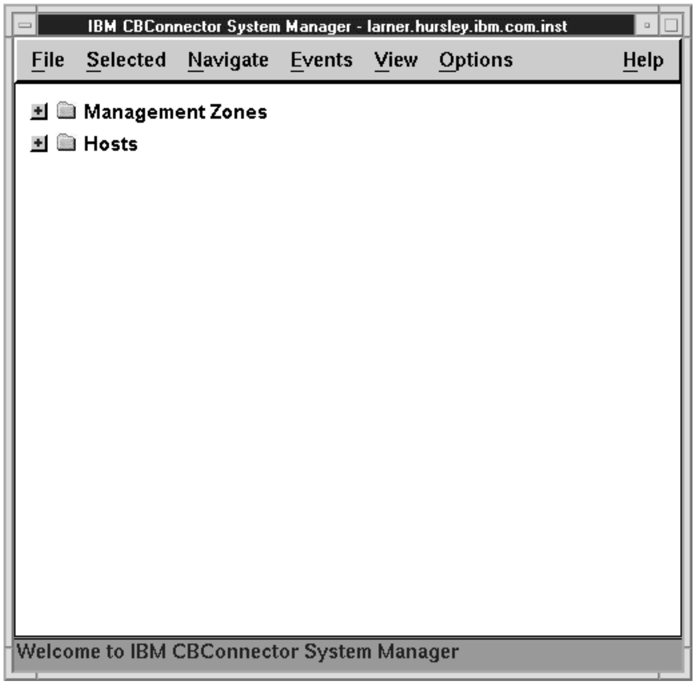
```
bhgeui '$eui' smhost
```

where *smhost* is the name of the host on which the System Manager is running; for example,

```
bhgeui '$eui' smhost.hursley.ibm.com
```

When you start the System Manager user interface, the following are displayed:

1. The Component Broker copyright message window, briefly.
2. The Information Controller window, as shown in Figure 19 on page 59, which displays the **Home** view of the system management object network. This window is the main window for user interaction with the System Manager.

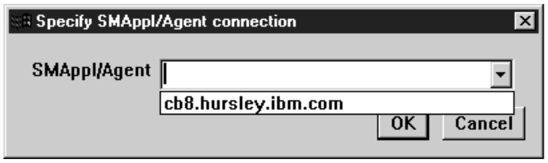


Welcome to IBM CBConnector System Manager

*Figure 19. The Information Controller window on AIX.*
*Showing the Home view of the system management object network.*

**Notes:**

1. If the System Manager user interface cannot connect to the System Manager automatically, it will prompt you for the full hostname of the System Manager (or SM Agent) that you want to connect the user interface to.



SMAppl/Agent: cb8.hursley.ibm.com

2. If you want Tivoli to be able to start the Component Broker System Manager user interface, you must “Install and Configure the Component Broker plus module for Tivoli” on page 47 and “Install a System Manager User Interface” on page 51 on the Tivoli host.

### Related Concepts

“System Manager User Interface” on page 12
“Appendix A. Features of the User Interface” on page 433
Installing Component Broker System Management Components
“Operating CBConnector System Management Components” on page 431

### Related Tasks

“Specify Colors and Fonts for the System Manager User Interface on AIX” on page 61
“Start the System Manager User Interface on Windows NT” on page 56
“Install a System Manager User Interface” on page 51
“Install and Configure the Component Broker plus module for Tivoli” on page 47
“Chapter 2. Use the System Manager User Interface” on page 55

## Exit the User Interface

To exit the user interface, you close the last Information Controller window by one of the following actions:

- Click the **X** button in the top-right corner of the window
- Double-click the icon in the top-left corner of the window
- Click **File - Exit Application**.

This displays a dialog box for you to confirm that you want to exit the user interface. To exit the user interface, click **Yes.** This closes the last Information Controller window and all its related windows (for example; Action Console and Object Editor windows). Otherwise, to continue with the user interface, click **No**.

If you use any of these actions when more than one Information Controller is displayed, only the one Information Controller is closed (along with all its related windows).

**Related Concepts**

"System Manager User Interface" on page 12
"Appendix A. Features of the User Interface" on page 433
"Chapter 2. Use the System Manager User Interface" on page 55

**Related Tasks**

"Start the System Manager User Interface on Windows NT" on page 56
"Start the System Manager User Interface on AIX" on page 58

## Change the Appearance of the SM User Interface

WIN The appearance of the SM user interface is controlled by the properties of the Windows NT desktop. To change the appearance of the SM user interface, use the **Display properties** window to change the settings; for example, in the **Item Size and Color** and **Font Size and Color** areas.

WIN **Tip** If you change the settings for the SM user interface, you can save them as a *scheme*. You can easily restore the settings later by selecting the scheme from the Windows NT Scheme list.

AIX You can specify your own fonts and colors for the SM user interface in your .Xdefaults file. **Note:** This is particularly important if you are using DBCS characters. For more information, see "Specify Colors and Fonts for the System Manager User Interface on AIX" on page 61.

**Related Concepts**

"Appendix A. Features of the User Interface" on page 433

**Related Tasks**

"Specify Colors and Fonts for the System Manager User Interface on AIX" on page 61
"Start the System Manager User Interface on Windows NT" on page 56

“Start the System Manager User Interface on AIX” on page 58
“Display Objects” on page 64
“Chapter 2. Use the System Manager User Interface” on page 55

## Specify Colors and Fonts for the System Manager User Interface on AIX

AIX Use this procedure to specify your own colors and fonts to be used by the System Manager User Interface on AIX.

**Note:** This is particularly important if you are using DBCS characters.

If you are not sure if appropriate colors and fonts are being used, “Start the System Manager User Interface on AIX” on page 58 then check the fonts and colors displayed. If they are not appropriate, specify your own fonts and colors as described in this topic, then restart the user interface.

For more information about specifying your own colors and fonts, see

- ″Specifying colors for the user interface″ (page 61)
- ″Specifying fonts for the user interface″ (page 61)

### Specifying colors for the user interface

The System Manager user interface first looks for any colors defined for it in the .Xdefaults file. If colors are not specified there, the default colors are loaded.

You can specify the colors for the System Manager user interface on AIX, by the **bhgeui*background** and **bhgeui*foreground** parameters in your .Xdefaults file. For example:

```
bhgeui*background: white
bhgeui*foreground: black
```

### Specifying fonts for the user interface

The fonts that the System Manager user interface can use depend on what you have installed on your AIX node. If the fonts in windows look strange, check that you are using appropriate fonts. If needed, you can specify your own fonts in your .Xdefaults file.

The System Manager user interface first looks for any fonts defined for it in the .Xdefaults file. If fonts are not specified there, the default fonts are loaded depending on the locale. These default fonts may not be available on your X server as each X server does not come with standard fonts. In this case, the default ″fixed″ font will be used. So it is better to set the fonts for the System Manager user interface in your .Xdefaults file.

Use the command **xset q** to display a status including the **font path** directories. (You can change the path using xset -fp or xset +fp). Look at the fonts.dir in each of these directories to see which fonts are available. The font definitions follow the following syntax:

```
-founder-facename-weight_name-slant-...-pointSize ...-m(p or c)..-codepage
```

You can specify the fonts for the System Manager user interface on the **bhgeui*fontList** parameter in your .Xdefaults file. For example:

```
bhgeui*fontlist:  -ibm—medium-b-medium—16-10-100-c-90-ibm-850
bhgeui*fontlist: Rom12.500
bhgeui*fontlist:IBM_JPN17;romankn17;kanji17;kana17:
```

**Note:** This is particularly important for DBCS users. If you have a bhgeui*fontlist resource in your .Xdefaults file that is based on SBCS, the System Manager user interface will not display DBCS text properly.

For DBCS you should set your own fonts in your .Xdefaults file; for example:

```
bhgeui*fontlist:IBM_JPN17;romankn17;kanji17;kana17:
```

### Related Concepts

"System Manager User Interface" on page 12
"Appendix A. Features of the User Interface" on page 433
"Operating CBConnector System Management Components" on page 431

### Related Tasks

"Start the System Manager User Interface on AIX" on page 58
"Chapter 2. Use the System Manager User Interface" on page 55

## Control Which Objects are Displayed

You can use a *user-level setting* to restrict the range of objects that a user can see and therefore act on. You can use an *object-level filter* to restrict further the range of objects displayed, *within the scope of the user-level setting*. Each Information Controller window has its own user-level setting and object-level filter.

To change the user-level setting, click one of the following options under **View - User Level**, listed from least-restrictive to most-restrictive levels:

**Basic**

This displays enough model objects for most applications to be configured and run, and enough Image objects for servers and applications to be started and stopped, and statistics gathered.

**Advanced**

This displays everything in the Basic view and extra model objects to allow tuning, and more Image objects to allow detailed statistics to be gathered.

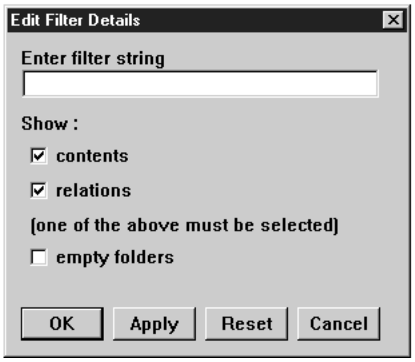
**Expert**

This displays everything in the Advanced view and all Image and Install objects; enough for problem determination.

New users are recommended to use the most restrictive user-level setting that meets their needs.

To change the object-level filter, use the **Edit Filter Details** window, by completing the following steps:

1. Open the Edit Filter Details window; for example, by selecting the filter icon of the Tool bar. This displays the following window:



Edit Filter Details

Enter filter string

Show :

☑ contents

☑ relations

(one of the above must be selected)

□ empty folders

2. Specify the filter details as:
   - A case-sensitive ASCII filter string, to be applied to all objects in the View panel. The filter string is applied to the names of objects displayed in the View panel, and can use the wild-card characters:

     * (asterisk) to match several characters

     ? (question mark) to match a single character

     For example, assume that the View panel displays the servers servhth1, servhth2, servith1, and SERVHTH3. The following filters would have the effect described:
     - **servht*** would result in only the servers servhth1 and servhth2 being displayed.
     - **serv?t*** would result in only the servers servhth1, servhth2, and servith1 being displayed.
     - **SERV*** would result in only the server SERVHTH3 being displayed.
   - Select whether the contents, relations, or both, of the current object are to be displayed in the View panel. To do this, select one or both of the tick boxes:

     **Contents**
     This toggles between showing and hiding the folders of objects contained in the current object.

     **Relations**
     This toggles between showing and hiding the folders of relationships contained in the current object.
   - Select whether empty folders are to be displayed in the View panel. To do this, select the **Show Empty Folders** tick box. This tick box toggles between displaying and hiding empty folders.
3. Select the **OK** button to apply the filter and close the Edit Filter Details window.

   Alternatively, to cancel any changes to the filter and close the window, select the **Cancel** button.

**Note:** The specified filter and object types to be displayed apply only to the View panel of the associated Information Controller window, and remain in force for all objects until reset.

### Related Concepts

"User-Level Settings and Object-Level Filters" on page 438
"System Management Objects used to Define your Enterprise" on page 42
"System Management Objects used to Operate your Enterprise" on page 45

### Related Tasks

"Display Objects"
"Act on Objects" on page 66
"Drag and Drop Objects" on page 69

## Display Objects

To be able to act on an object directly, you must first display it in the View panel of the Information Controller window. To display an object, you can use one or more of the following actions:

- If the object is in your hotlist, display the Hotlist window (click the Hotlist icon ▦ of the Tool bar), then double-click the left mouse button on the required object.
- If the object is in your session history (was displayed during this session with the Information Controller window), display the History window (click the History icon ⧉ of the Tool bar), then double-click the left mouse button on the required object.
- If the object was displayed within a reasonable number of steps on the route to the current object, use the **Backward** action; for example, by clicking the Backward icon ⧉ of the Tool bar.
- If the object has a *relationship* with the current object, open the appropriate relationship to display the required object.
- If the object is *lower down* the object network than the current object, the easiest way to find it is to use **Tree view** (select from the **View** menu bar choice), and expand appropriate branches until the required object is displayed. This preserves the tree structure.

  Alternatively, you can **open** appropriate objects until the required object is displayed. This replaces the information in the View panel with the contents of the object opened.
- If the object is *higher up* the object network than the current object, you can use the following actions to find the object:
  - Click **Navigate - Up** to display the next higher level up the object network
  - Click the Home icon 🏠 of the Tool bar) to display the Home view of the object network. You can then expand the Home view as a tree structure or open appropriate objects to display the required object.

When you have displayed the required object, you can act on it using its pop-up menu or select it to use the menu bar choices or Tool bar.

### Related Concepts

"Using The Mouse And Keyboard" on page 436
"Generic Actions" on page 69

### Related Tasks

"Create Objects" on page 65
"Select and Deselect Objects" on page 65
"Act on Objects" on page 66
"Drag and Drop Objects" on page 69

## Create Objects

You can create a new object simply by inserting one into the folder for the object class, using either of the following methods:

- On the pop-up menu of the object that contains the folder for the new object, click **New**, then select the class of object to be created. For example, on the pop-up menu of a Configuration click **New - Server Group**, to create a new Server Group. (You are prompted to specify an appropriate name for the new object.)
- Insert the new object directly into its folder, by completing the following steps:
  1. “Display Objects” on page 64.
  2. On the folder’s pop-up menu, click **Insert**.
  3. Specify an appropriate name for the new object
  4. To create the object, click the **OK** button.

     To cancel the action, click the **Cancel** button.

Both methods display a dialog box for you to identify the new object. If you type a valid name, the new object is created in the folder. To display the object, and perhaps act on it, expand the folder.

You can use both methods to create new objects for object classes allowed by the current user-level setting only. If you need to create an object for which the class is not displayed for the **New** action or the object’s folder cannot be displayed, “Control Which Objects are Displayed” on page 62 to a less-restrictive value.

**Related Concepts**

“System Management Objects used to Define your Enterprise” on page 42
“Using The Mouse And Keyboard” on page 436

**Related Tasks**

“Display Objects” on page 64
“Act on Objects” on page 66
“Drag and Drop Objects” on page 69

## Select and Deselect Objects

If you select an object, you can use more than just its pop-up menu to act on the object. Furthermore, to act on several objects at the same time, you must first select all those objects. For example, you can select several Application Models, then use the **Edit** action from the **Selected** menu bar choice to edit all the selected objects. To indicate that an object is selected, it is visibly highlighted. To select objects, you can:

- Select one object
- Select several objects individually
- Add objects to a group of selected objects
- Select all objects in a panel

There are several ways to select objects using the mouse and keyboard, detailed in the **Keys Help** choice of the online help. (To display the help, click the **Help** menu bar choice.) The most common ways are:

- To select one object, click on the object
- To select several adjacent objects, in List view complete the following steps:
  1. Click the first object
  2. Press and hold down the Shift key
  3. Click the last object

Usually you do not need to explicitly deselect objects, because objects are deselected automatically when you select another object, unless special actions are taken. However, there are occasions when you would want to deselect objects; for example, to deselect objects in the View panel so that you can act on the current object (that contains them) without having to display that object in the View panel.

As for selecting objects, there are several ways to deselect objects. The most common way is click on the currently selected object. When an object is deselected, its highlight is removed.

To deselect all objects displayed in List view, click on the background of the View panel.

**Related Concepts**

"System Management Objects used to Define your Enterprise" on page 42
"System Management Objects used to Operate your Enterprise" on page 45
"Using The Mouse And Keyboard" on page 436

**Related Tasks**

"Display Objects" on page 64
"Create Objects" on page 65
"Act on Objects"
"Act on Several Selected Objects" on page 68
"Drag and Drop Objects" on page 69

## Act on Objects

You can use the menu bar and pop-up menus to act on the current object and other objects displayed.

Objects, relationships, and attributes are protected against change by anyone other than with the appropriate user-level setting. Under normal conditions, you can act on all the objects that you need within the **Basic**, **Advanced**, and **Expert** user-level settings. If you cannot act on an object with your current user-level, consider "Control Which Objects are Displayed" on page 62 and, when finished, reset the user-level setting back to a suitable more-restrictive value.

There are two main categories of actions that you can use on objects:

1. **Simple actions** require no data input. In task descriptions, these are referred to only by the action name and identify the object that they are used on; for example, "**Start** the server called 'Sample Server 1'".
2. **Dialog actions** present a dialog box for you type or select data, then click the box's **OK** button to complete the action. In task descriptions, these are referred to by the action name, the data to be provided or selected, and identify the object that they are used on; for example, "**Insert** a configuration called **Sample**

**Configuration 2** into the Configurations folder". The action description implicitly assumes that you click the **OK** button to complete the action. (If you want to cancel a dialog action, you cick the **Cancel** action.)

This simplification of action descriptions makes it easier for you to follow the actions.

You can choose how you select an action, but generally the most convenient way to act on a single object is to use the pop-up menu of the object. *The pop-up menu can be used to act on only the one object that is being pointed at.* To display the pop-up menu:

1. Move the mouse to point at the object
2. Press and hold the right mouse button
3. Move the mouse to point at the required action
4. Release the right mouse button

**Note:** You do not need to first select an object to use its pop-up menu.

Alternatively, you can select an object by clicking on it, then select an action from the available menu-bar choices or the Tool bar.

The pop-up menu for an object and the **Selected** menu bar choice display all the actions that can be used directly on the object. The menu displays generic actions that can be used on most classes of objects and other actions that are specific to the class of object.

In the pop-up menu for an object, only those actions that can be used in the current *state* of the object can be selected. As an example, the choices on the pop-up menu for a Server Image are shown in Figure 20.

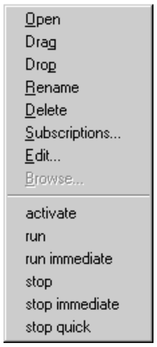


*Figure 20. The Pop-up Menu for a Server Image*

To act on several selected objects, you use the **Selected** menu bar choice, which displays only those actions that can be used on *all* the selected objects. For more information, see “Act on Several Selected Objects” on page 68.

For more information about the actions that you can take on an object, use the online help for the object or see other task topics.

### Related Concepts

“Help Information” on page 449
“Appendix A. Features of the User Interface” on page 433
“Generic Actions” on page 69

### Related Tasks

"Chapter 2. Use the System Manager User Interface" on page 55
"Display Objects" on page 64
"Act on Several Selected Objects"
"Drag and Drop Objects" on page 69

## Act on Several Selected Objects

To act on several objects at the same time you must first select those objects *in List view*. (You can select more than one object in List view only.) You can then use one action from the **Selected** menu bar choice to affect all the selected objects. Such actions are identified by **Selected -** *action*; for example, "Click **Selected - Drag**".

The **Selected** menu bar choice displays only those actions that can be used on *all* the selected objects.

For example, to configure several Application Models onto a Server Model, you take the following steps:

1. Switch the view type to **List view**
2. Select several Application Models
3. Click **Selected - Drag**, to drag all the selected Application Models
4. On a Server Model's pop-up menu, click **Configure Applications**, to configure all the Application Models on the Server Model

To add one or more objects to a group of selected objects:

1. Move the mouse to point at an object
2. Press and hold down the Ctrl key
3. Press the left mouse button
4. If you want, release the Ctrl key
5. Navigate to another object
6. Press and hold down the Ctrl key
7. Press the left mouse button to add the object to the selected group

Repeat steps 5 to 7 for each object you want to add to the group of selected objects.

To deselect one of several selected objects:

1. Move the mouse to point at the object.
2. Press and hold down the Ctrl key.
3. Press the left mouse button.

You can repeat this action to deselect other objects.

### Related Concepts

"System Management Objects used to Define your Enterprise" on page 42
"System Management Objects used to Operate your Enterprise" on page 45
"Generic Actions" on page 69

### Related Tasks

“Display Objects” on page 64
“Act on Objects” on page 66
“Drag and Drop Objects”

## Generic Actions

You can use the actions listed in the table Generic Actions (page 69) to perform general actions on most objects; for example, to **open** an object or **edit** it.

The actions available for an object depend on the class of object and the current user-level setting. If an action is not available, the menu option is displayed in grey, and cannot be selected.

*Table 1.* **Generic actions**

| | |
|---|---|
| **Open** | Make this object the current object in the Information Controller window and display the folders and objects that it contains in the View panel. |
| **Drag** | Drag this object to enable it to be dropped onto other objects or relationship fields. |
| ″drop″ | Drop dragged objects onto this object or relationship field to perform context-sensitive actions. The ″Drop″ action is displayed as appropriate context-sensitive menu options; for example, **Configure Application** on a Server Group. |
| **Edit** | Open the Object Editor window for this object, to display or change its attributes. |
| **Rename** | Rename this object. |
| **Delete** | Delete this object. |
| **Subscriptions** | Display the Object Subscriptions window for this object, to display and act on this object’s event subscriptions. |
| **Insert**<br>(For folders only) | Insert (create) a new object into this folder. |
| **New** | Insert (create) a new object within this object; for example, **New - Server Group** on a Configuration, to create a new Server Group within that Configuration. |

### Related Concepts

“Help Information” on page 449
“Appendix A. Features of the User Interface” on page 433

### Related Tasks

“Display Objects” on page 64
“Act on Objects” on page 66
“Drag and Drop Objects”

## Drag and Drop Objects

You can act on objects by dragging one object and dropping it onto either of the following:

- “Drag and Drop Objects onto Objects” on page 70
- “Drag and Drop Objects onto Relationships” on page 71

Drag and drop tasks involve two actions; clicking **Drag** for the object to be dragged, and clicking an appropriate context-sensitive action for the target object or relationship. The **Drag** action causes context-sensitive actions to be added to the pop-up menus of other objects and relationships.

**Related Concepts**

“System Management Objects used to Define your Enterprise” on page 42
“System Management Objects used to Operate your Enterprise” on page 45

**Related Tasks**

“Drag and Drop Objects onto Objects”
“Drag and Drop Objects onto Relationships” on page 71
“Copy Attribute Strings” on page 72
“Display Objects” on page 64
“Act on Objects” on page 66
“Edit Objects” on page 72

## Drag and Drop Objects onto Objects

You can use the System Manager user interface to act on two objects by dragging one object and ″dropping″ it onto the other object. The ″drop″ action is identified by a context-sensitive label that indicates the effect of the action; for example, if you drag an Application then display the pop-up menu of a Server Group, the menu shows the context-sensitive action **Configure Application**. In many cases, such actions are to create relationships between the two objects. The drag and drop actions between objects are given in the descriptions of the object classes, in Object Reference.

You can also create other relationships between objects by dragging an object and dropping it onto a relationship folder of another object, as described in “Drag and Drop Objects onto Relationships” on page 71.

To drag an object and drop it onto another object, complete the following steps:

1. On the pop-up menu of the object to be dragged, click **Drag**
2. Display the target object. If it is not displayed, set the filter (and if needed, user-level setting) to display it.
3. On the target object’s pop-up menu, click the appropriate context-sensitive action

Context-sensitive actions are displayed for only those objects that the dragged object can be dropped onto.

**Notes:**

1. To drag one object, you can click either **Drag** on the object’s pop-up menu or **Selected - Drag**. To drag several objects with one action, you must click **Selected - Drag**.
2. To drop onto one object, you can click a context-sensitive action on either the object’s pop-up menu or the **Selected** menu-bar choice. To drop onto several objects with one action, you must click a context-sensitive action on the **Selected** menu-bar choice.

3. When you drag objects, they can be dropped by any number of subsequent context-sensitive actions. When you select another **Drag** action, the newly dragged objects become the current objects used by subsequent context-sensitive actions.

**Related Concepts**

"System Management Objects used to Define your Enterprise" on page 42
"System Management Objects used to Operate your Enterprise" on page 45

**Related Tasks**

"Drag and Drop Objects onto Relationships"
"Copy Attribute Strings" on page 72
"Drag and Drop Objects" on page 69
"Display Objects" on page 64
"Act on Objects" on page 66
"Edit Objects" on page 72

## Drag and Drop Objects onto Relationships

You can Drag an object and drop it onto a target relationship folder of another object to create that relationship between the two objects.

To Drag and drop objects onto relationships, complete the following steps:

1. On the object's pop-up menu, click **Drag**.
2. Expand the target object to be related. If the required relationship folder is not displayed, set the filter (and if needed, user-level setting) to display it.
3. On the pop-up menu of the appropriate relationship folder, click **Create Relationship**.

If any of the Dragged objects cannot be dropped onto the selected target relationship, the System Manager user interface displays a message dialog box to warn you. You can still drop the Dragged objects onto other objects or relationships.

**Notes:**

1. To Drag one object, you can click either **Drag** on the object's pop-up menu or **Selected - Drag**. To Drag several objects with one action, you must click **Selected - Drag**.
2. When you Drag objects, they can be dropped by any number of subsequent **Create Relationship** actions. When you select another **Drag** action, the newly Dragged objects become the current objects used by subsequent **Create Relationship** actions.

**Related Concepts**

"System Management Objects used to Define your Enterprise" on page 42
"System Management Objects used to Operate your Enterprise" on page 45

**Related Tasks**

"Display Objects" on page 64
"Act on Objects" on page 66
"Drag and Drop Objects" on page 69

"Drag and Drop Objects onto Objects" on page 70
"Copy Attribute Strings"
"Edit Objects"

### Copy Attribute Strings

To copy an object's attribute string to another attribute, use standard copy and paste functions. The two attributes can be for different objects. For example, complete the following steps:

1. In one attribute field, select the string that you want to copy
2. Press **Ctrl-Insert** to copy the string
3. Click the attribute field where you want to copy the information
4. Press **Shift-Insert** to paste the string

If the copied attribute string is not valid for the attribute to which the string is copied, a message dialog box is displayed to warn you. You can still paste the copied string onto another attribute. The copied attribute string is kept as the current string to be pasted, until you copy another attribute value.

**Related Tasks**

"Display Objects" on page 64
"Act on Objects" on page 66
"Drag and Drop Objects" on page 69
"Drag and Drop Objects onto Objects" on page 70
"Drag and Drop Objects onto Relationships" on page 71
"Edit Objects"

## Edit Objects

Each object that you can act on through the user interface has attributes that you can edit.

To display and edit the attributes of an object, complete the following steps:

1. Display the Object Editor window for the object; for example, by clicking **Edit** on the object's pop-up menu. This displays the Object Editor as a notebook with a title page and one or more attribute pages for the object.

   All objects have the following pages:

   **Tab** **Description**

   **<object name>**
   Displays the full location of the object within the CBConnector System Management object network.

   **Main** Displays the most important attributes of the object.

   Some objects also have pages to display the performance statistics of the object and other pages specific to the class of object.
2. Select the type of view that you want to use to display attributes, by clicking on either of the following menu choices:

   **View - View Type - Full Panel View**
   Display the list of attributes each with an entry field (the default)

**View - View Type - Listbox View**
Display the list of attributes with one shared entry field

3. To change the value for an attribute, either type a new value in the value field or select a value from the field's pull-down box:
   - In **Full Panel View**, use the attribute's value field.
   - In **Listbox View**, select the attribute to display the current value in the (one) value field, then use that field to change the value.

   **Note:** When you change any attribute, the attribute is marked by changing its color and adding a "greater than" character (>) as a prefix. This is to help you identify attributes that you have changed. If the change is not valid, an error message box dialog is displayed, and the reason for the error is displayed in the status field.
4. When you have made all the changes that you want, choose whether to apply or discard the changes by one of the following actions:
   - To apply the changes and close the Object Editor window, click the **OK** button.
   - To apply any changes and remain in the Object Editor window, click the **Apply** button.
   - To discard any changes and return to the current Information Controller window, click the **Cancel** button.

**Related Concepts**

"Appendix A. Features of the User Interface" on page 433
"Generic Actions" on page 69

**Related Tasks**

"Display Objects" on page 64
"Act on Objects" on page 66
Display The Object Editor

## Display Help Information

To display context-sensitive help information about a window element or object, select the element or object then press the F1 key.

To display other help information, select the **Help** menu bar choice.

**Related Concepts**

"Help Information" on page 449
"Appendix A. Features of the User Interface" on page 433
"Generic Actions" on page 69

**Related Tasks**

"Display Objects" on page 64
"Act on Objects" on page 66
"Drag and Drop Objects" on page 69

## Example: Explore the User Interface

This example guides you round the main parts of the System Manager user interface. It is based on the Windows NT version of the user interface, and assumes that you have just started it asdescribed in “Start the System Manager User Interface on Windows NT” on page 56. This displays the Information Controller window that shows the Home view of the object network in its View panel, as shown in Figure 21.

The example also describes some significant objects to clarify actions that are taken on those objects later in the example. It does not describe all objects that you can see, nor all parts of the user interface; such details are given in other information topics.

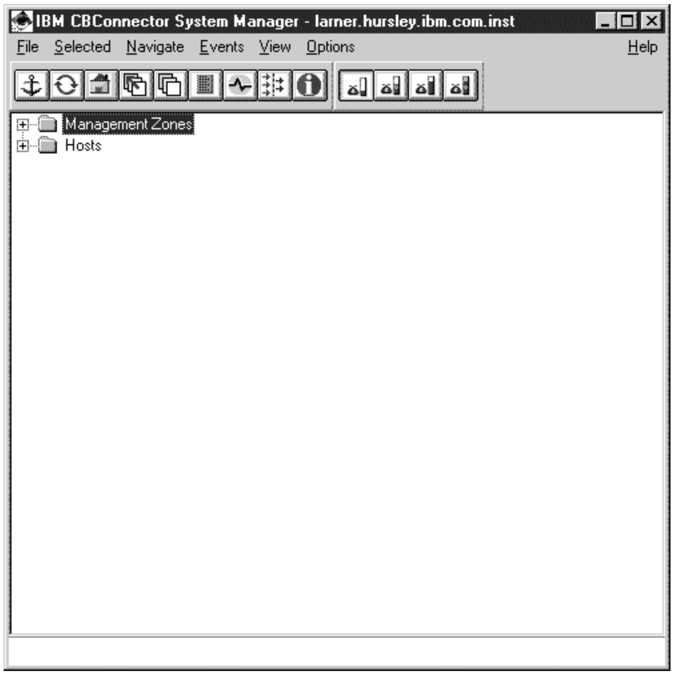


*Figure 21. The Information Controller window.*
*Showing the Home view of the CBConnector System Management object network.*

The main area of information in this window is called the **View panel**. The View panel displays icons or details of the objects that you have selected.

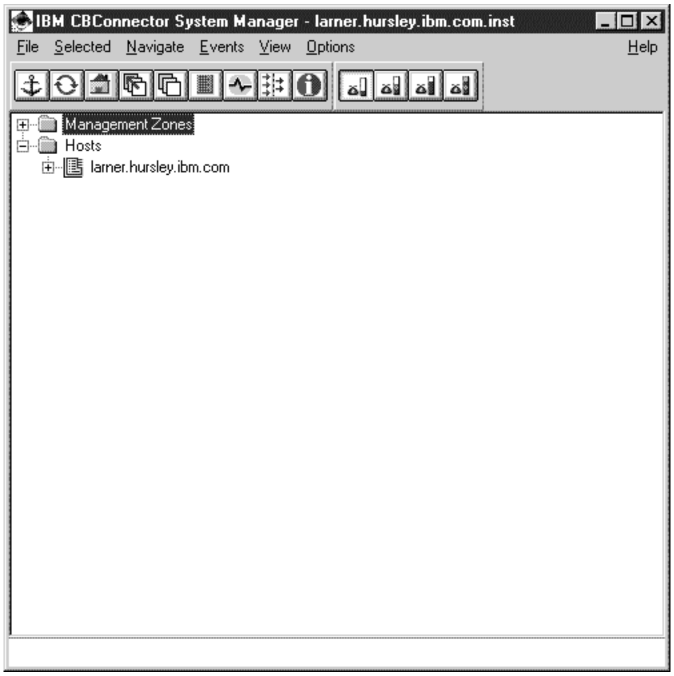
In this example, the View panel displays the *Home view* of the CBConnector System Management object network as the top level of a tree structure. It displays the following folders:

**Management Zones**
The folder that contains objects that define the model world of the entire managed enterprise. Each *Management zone* contains objects and relationships that define servers, clients, applications, and other objects for all or part of the enterprise.

**Hosts** The folder that contains objects that represent the defined view of managed hosts. Each *Host Model* in this folder contains relationships that point to *model objects* (in a management zone) that represent servers, clients, applications, and other objects that are defined for that host.

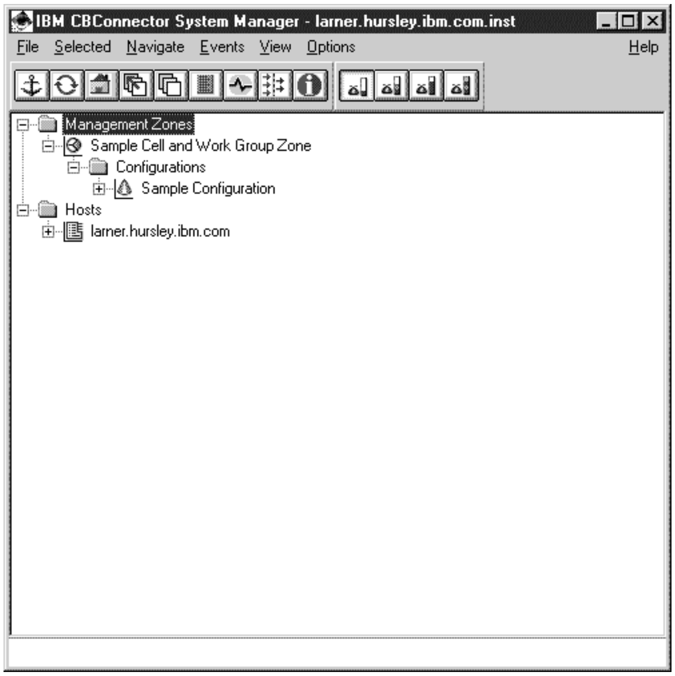
The number and types of objects displayed is initially restricted, because the *user-level* is set to the minimum, **basic**, and the *filter* is set to hide empty folders. The user-level setting and filter enable you to control the view of objects that you can act on, so that you can better focus on what you need to do. Later in this example, you are shown how changing the user-level setting and filter changed what is displayed in the View panel.

1. To display the contents of the Hosts folder, click on the + symbol to the left of that folder object.


IBM CBConnector System Manager - larner.hursley.ibm.com.inst
File Selected Navigate Events View Options Help
Management Zones
Hosts
larner.hursley.ibm.com

   Note how the contents of the folder are displayed to the right of the Host Image folder object and the + symbol changes to a minus symbol. To display the contents of any object in a task description, you would be instructed to **expand** the Host Image folder.

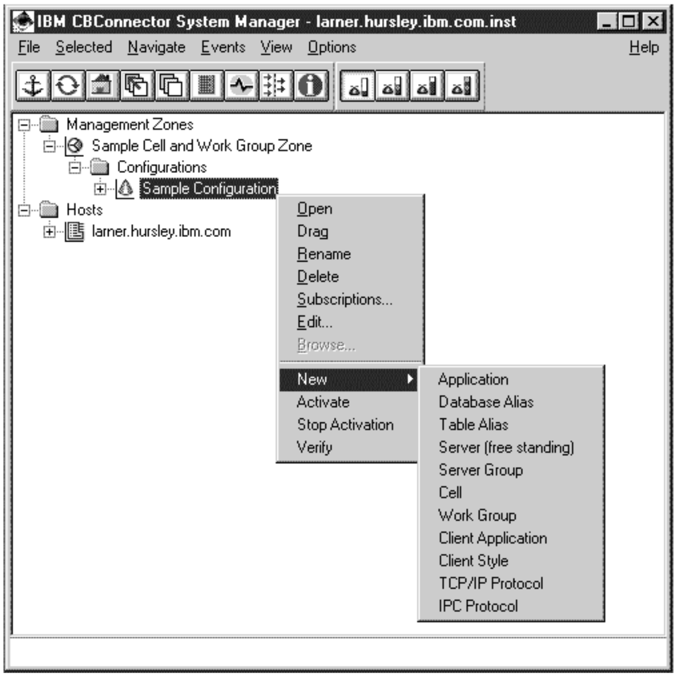
2. You can click on the minus (-) symbol next to an object to hide its contents. Try this on the Hosts folder.
3. Expand the Hosts folder and the Management Zones folder. Note that Management Zones folder. contains a sample Management Zone, which is created automatically when you install CBConnector System Management. (You may see more than one sample Management Zone, depending on the options that you selected when you installed CBConnector System Management, and on what you have done with CBConnector System Management since.)
4. Expand the sample Management zone then expand its **Configurations** folder.


IBM CBConnector System Manager - larner.hursley.ibm.com.inst
File Selected Navigate Events View Options Help
Management Zones
Sample Cell and Work Group Zone
Configurations
Sample Configuration
Hosts
larner.hursley.ibm.com

Note that the Configurations folder contains a sample Configuration, which is created automatically when you install CBConnector System Management. (You may see more than one sample Configuration, depending on the options that you selected when you installed CBConnector System Management, and on what you have done with CBConnector System Management since.)

The usual way to act on an object in the View panel is through the object's pop-up menu.

5. Display the pop-up menu of the sample Configuration by clicking the right mouse button on that object.



Note the range of actions that can be used generally on all objects. These common actions, **Open** down to **Browse**, are separated by a line from the actions specific to the class of object selected. In this case, the Configuration-specific actions are **New** down to **Verify**.

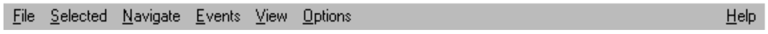
Note the classes of new objects that you can create within the Configuration, by using the **New** menu-choice. Any new objects created are added to their currently empty class folders, which would then be displayed. The class folders are currently hidden, because the filter is set to hide empty folders.

6. To hide the pop-up menu again, click away from the pop-up menu such as on the window background.

The Information Controller window also displays the following two main areas for acting on the contents of the View panel:

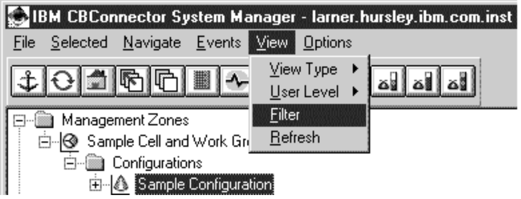
- Menu bar
- Tool bar

The menu bar



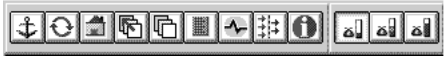
displays a number of choices of drop-down menus. In task descriptions, each option on a drop-down menu is identified by the menu choice name and the option name; for example, **View - User Level**.

1. Try clicking on the **View - User Level** option.



The **View - User Level** option enables you to change the user-level setting for this SM user interface. The user-level setting controls what can be displayed for the user of an SM user interface, and therefore controls what that user can act on. The levels displayed are for different types of users, listed from the most-restrictive to least-restrictive levels: The default level, **basic**, displays enough model objects for most applications to be configured and run, and enough image objects for servers and applications to be started and stopped, and statistics gathered. To clear the menu, click on the window background.

The tool bar

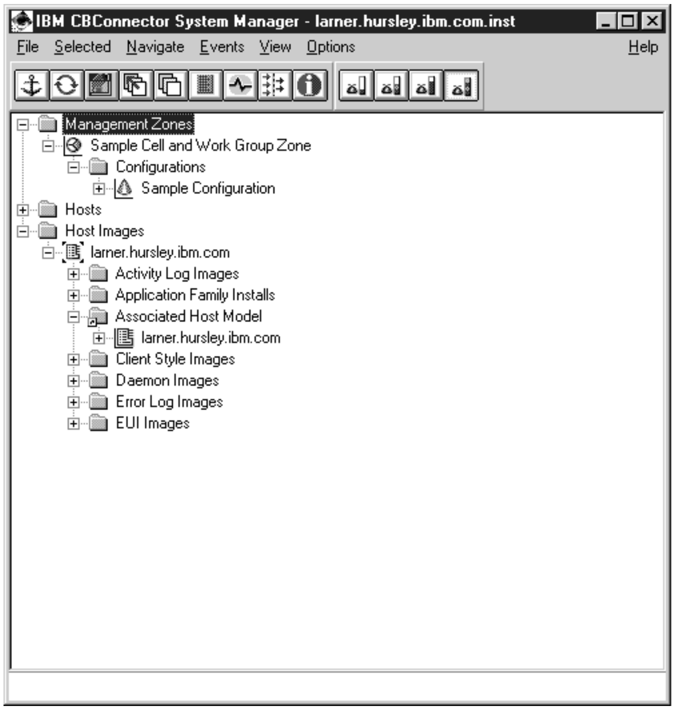


displays a number of icons for system management tools and functions, to provide a more convenient way to use them than through the menu bar. For example, move the mouse pointer to cross the user-level setting icons of the Tool bar. Note the bubble-help descriptions of the icons that are displayed as the mouse pointer crosses over them.

2. Click on the **Set User Level to Expert** icon .

   Note that you can now see the **Host Images** folder.

3. Expand the **Host Images** folder. Note that there is a Host Image for the Host Model in the **Hosts** folder. If you expand the Host Image, you can see Images that represent the objects being managed on the host and the application families installed on that host. You can also see the *Associated Host Model* relationship that relates the Host Image to its Host Model.



**Related Concepts**

"System Manager User Interface" on page 12
"Appendix A. Features of the User Interface" on page 433

**Related Tasks**

"Example: Change the Values of Attributes" on page 198
"Start or Stop the System Manager User Interface" on page 55
"Exit the User Interface" on page 60

## Drag and Drop Actions

This topic summarizes the drag and drop actions that you can use to configure model objects. The drag and drop actions can be performed in either direction, as listed, by using the **Drag** action on one object (the *dragged object*) and a context-sensitive ″drop″ action on another object (the *target object*).

The actions are listed under the following categories:

**Cells and Workgroups**
Configure hosts as members of cells and workgroups, configure name servers for cells and workgroups, and relate cells to workgroups

**Client styles and applications**
Configure applications onto client styles, client styles onto their hosts, and the bootstrap hosts for client styles.

**Factory finders**
Configure factory finders onto the location that they use.

**Location Chains**
Configure an ordered list of location objects by linking an Ordered Location to a chain of Location Chains, and linking each Location Chain to a location object.

**Profiles and profile classes**
Configure profiles onto profile classes and managed object classes onto the profile class that they use.

**Servers (freestanding)**
Configure applications onto freestanding servers, and servers onto their hosts

**Server groups**
Configure servers as members of server groups, applications onto server groups, server groups onto their controlling hosts, and member servers onto their hosts

**Workload management**
Configure bind policies onto policy groups, policy groups onto their containers, and C++ classes onto the policies that they implement

**Protocols**
Configure protocols for use on hosts

**Remote Name Contexts**
Configure remote name contexts for use on hosts

**Objects provided by applications**
Configure applications onto the objects that they provide

### Protocols:

#### Configure a TCP/IP protocol onto a host so that the protocol can be used on that host.

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| TCPIP Protocol | Host | Configure TCP/IP Protocol |
| Host | TCPIP Protocol | Configure Host |

#### Configure an IPC protocol onto a host so that the protocol can be used on that host.

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| IPC Protocol | Host | Configure IPC Protocol |
| Host | IPC Protocol | Configure Host |

### Cells and Work Groups:

#### Configure a host as a member of a cell.

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Host | Cell | Configure Member Host |
| Cell | Host | Configure Cell Member |

#### Configure the preferred cell for a host.

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Host | Cell | Configure Preferring Host |

| | | |
|---|---|---|
| Cell | Host | Configure Preferred Cell |

**Configure the host that provides the name server for a cell. The Host’s name server also serves as the cell’s name server.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Host | Cell | Configure Serving Host |
| Cell | Host | Configure Served Cell |

**Configure a host as a member of a work group.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Host | Work Group | Configure Member Host |
| Work Group | Host | Configure Work Group Member |

**Configure the preferred work group for a host.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Host | Work Group | Configure Preferring Host |
| Work Group | Host | Configure Preferred Work Group |

**Configure the host that provides the name server for a work group. The Host’s name server also serves as the work group’s name server.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Host | Work Group | Configure Serving Host |
| Work Group | Host | Configure Served Work Group |

**Define that a cell and work group that share common hosts.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Work Group | Cell | Configure Associated Work Group |
| Cell | Work Group | Configure Associated Cell |

**Servers (free standing):**

**Configure an application for use on a freestanding server. When the configuration is activated, the application is made available on the server.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Application | Server (free standing) | Configure Application |
| Server (free standing) | Application | Configure Server |

**Configure a freestanding server onto the host on which the server is to run. When the configuration is activated, the server can be started on that host.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Server (free standing) | Host | Configure Server (free standing) |
| Host | Server (free standing) | Configure Host |

**Server Groups:**

**Configure an application onto a server group. When the configuration is activated, the application is made available on servers that are defined as members of that server group.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|

| | | |
|---|---|---|
| Application | Server Group | Configure Application |
| Server Group | Application | Configure Server Group |

**Configure a server group onto a host. This configures the server group as a *controlled server group* for workload management. When the configuration is activated, the SGCP and SGGW servers for that server group are created on that host.**

| **Dragged Object** | **Target Object** | **Action on Target Object** |
|---|---|---|
| Server Group | Host | Configure Managed Server Group |
| Host | Server Group | Configure Managing Host |

**Configure a member of a server group onto the host on which that server is to run. When the configuration is activated, the server can be started on that host.**

| **Dragged Object** | **Target Object** | **Action on Target Object** |
|---|---|---|
| Server (member of group) | Host | Configure Server (member of group) |
| Host | Server (member of group) | Configure Host |

**Workload management:**

**Configure a policy group onto a container that is to use it.**

| **Dragged Object** | **Target Object** | **Action on Target Object** |
|---|---|---|
| Container | Policy Group | Configure Container |
| Policy Group | Container | Configure Policy Group |

**Configure a bind policy as a member of a policy group.**

| **Dragged Object** | **Target Object** | **Action on Target Object** |
|---|---|---|
| Policy Group | Bind policy | Configure Policy Group |
| Bind policy | Policy Group | Configure Bind Policy |

**Configure a bind policy onto the C++ class that implements the policy.**

| **Dragged Object** | **Target Object** | **Action on Target Object** |
|---|---|---|
| Bind policy | C++ Class | Configure Bind Policy |
| C++ Class | Bind policy | Configure C++ Class |

**Profiles:**

**Configure a profile as a member of a profile class.**

| **Dragged Object** | **Target Object** | **Action on Target Object** |
|---|---|---|
| Profile | Profile Class | Configure Profile |
| Profile Class | Profile | Configure Profile Class |

**Configure a managed object class onto the profile class that it uses.**

| **Dragged Object** | **Target Object** | **Action on Target Object** |
|---|---|---|
| Profile Class | Managed Object Class | Configure Profile Class |
| Managed Object Class | Profile Class | Configure Managed Object Class |

**Factory Finders:**

**Configure a factory finder onto the location that it uses.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Factory Finder | Single Location or Ordered Location | Configure Factory Finder |
| Single Location or Ordered Location | Factory Finder | Configure Location |

**Location Chains:**

**Configure an ordered list of location objects by linking an Ordered Location to a the first Location Chain in a chain, linking each Location Chain to the next one in the chain, and linking each Location Chain to a location object (a Single Location or Ordered Location).**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Ordered Location | Location Chain | Configure Owning Ordered Location |
| Location Chain | Ordered Location | Configure First in Owned Chain |
| Location Chain | Location Chain | Configure Next in Chain |
| Location Chain | Location Chain | Configure Previous in Chain |
| Location Chain | Single Location or Ordered Location | Link to Chain |
| Single Location or Ordered Location | Location Chain | Link Location |

**Client Styles and Applications:**

**Configure a client application onto a style of client. When the configuration is activated, the application is made available on that client style.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Client Application | Client Style | Configure Client Application |
| Client Style | Client Application | Configure Client Style |

**Configure a client style onto a host. When the configuration is activated, the client style is made available on that host.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Client Style | Host | Configure Client Style |
| Host | Client Style | Configure Host |

**Configure a client style onto its bootstrap host. When the configuration is activated, the clients of the client style will use the host as their bootstrap host.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Client Style | Host | Configure Client Style for Bootstrap |
| Host | Client Style | Configure Host for Bootstrap |

**Remote Name Contexts:**

**Configure a remote name context onto a host so that the remote naming context binding can be bound into the name space for that host.**

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|

| | | |
|---|---|---|
| Remote Name Context | Host | Configure Remote Name Context |
| Host | Remote Name Context | Configure Host |

**Objects Provided by Applications:**

**Configure an object for use by an application on an application server. When the application is configured onto an application server, the object is configured for use on that server.**

To configure an object for use by its application on an application server, you use the **Configure Application** action, to create a *providing application* relationship. For example, to configure a factory finder for use by an application on its application servers, you can use the following drag and drop actions:

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Application | Factory Finder | Configure Application |
| Factory Finder | Application | Configure Factory Finder |

**Configure an object for use by an application on a host name server. When the application is configured onto an application server, the object is configured for use on the name server for the same host.**

To configure an object for use by its application on host name servers rather than on specific application servers, you use the **Configure Application (for host)** action, to create a *providing application for host* relationship. For example, to configure a factory finder for use on host name servers rather than on specific application servers, you can use the following drag and drop actions:

| Dragged Object | Target Object | Action on Target Object |
|---|---|---|
| Application | Factory Finder | Configure Application (for host) |
| Factory Finder | Application | Configure Factory Finder (for host) |

### Related Concepts

"The Model World" on page 39
"Generic Actions" on page 69

### Related Tasks

"Drag and Drop Objects" on page 69

# Chapter 3. Create a New Single-Host Enterprise

This topic describes how to create and configure a new Enterprise that uses one standalone host to support applications that run on application servers on that host.

*The standalone host configured in this topic is intended more for testing Component Broker and as a base for an application development host.* For a more realistic enterprise in which to deploy your application environment, see “Chapter 4. Create a New Multi-Host Enterprise” on page 101.

**Note:** The tasks used to configure a single-host enterprise are the same as the first steps for configuring a multi-host environment. At a later time, you can add more hosts to your single-host environment to expand the environment into a multi-host environment.

The tasks used are based on the following assumptions:

- You have already installed Component Broker onto the host to be configured, as described in the *Planning, Performance, and Installation Guide*. For an overview of installing a Component Broker single-host environment, see “Install a Component Broker Single-Host Environment” on page 88.
- You have not already activated a system management Configuration that defines the cell, work group, and serving hosts of your Component Broker host environment. If you have done this, and want to reorganize your host environment, see “Chapter 6. Administer your Host Environment” on page 167.

If you want to use the sample Management Zone provided with Component Broker, follow the steps given in the example, “Example: Use the Sample Configuration for a Single-Host Environment” on page 95 instead of those listed below.

If you are already familiar with using the System Manager user interface, and have a good plan of what you want your enterprise to be, you can skip the initial overview, and start by “Configure a Single-Host Environment” on page 90.

## Overview of Configuring a Single-host Enterprise

This topic describes things that you should consider when preparing to create a new standalone host, based on the general host shown in Figure 22 on page 86.

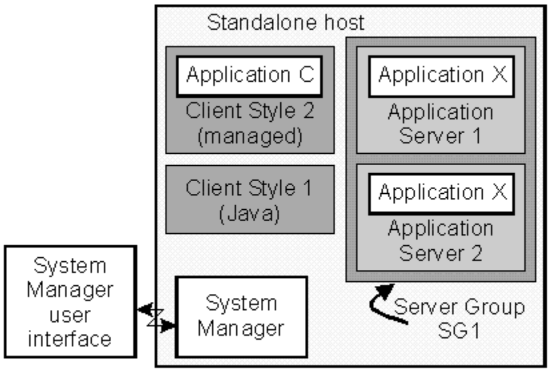


*Figure 22. A Single-Host Enterprise*

When preparing to create any enterprise, the first aim is to get a clear plan of what you want to create. The planning and creation of any enterprise can be simplified by considering the enterprise in two parts; the *host environment* and the *application environment*.

**The Host Environment**

When creating a new single-host Component Broker enterprise, you need to configure the host into the Component Broker cell and the minimum work group. Also, the hostneeds to be configured with a name server and TCP/IP protocol.

For more information about what you need to configure for the host environment, see “Plan what to Configure for a Single-host Environment” on page 87.

**The Application Environment**

The application environment comprises the clients, servers, and applications that run on the host managed by Component Broker. It also can include clients that run on workstations and other hosts that are not managed by Component Broker; for example, Java clients.

When creating any new enterprise, you need to configure the application environment into server groups, freestanding servers, and client styles. You also need to load applications into Component Broker and configure those applications as instructed by information provided with each application.

For more information about what you need to configure for an application environment, see “Plan what to Configure for a Single-Host Application Environment” on page 134.

If you want to proceed with configuring a new single-host enterprise, complete the following main tasks:

1. “Configure a Single-Host Environment” on page 90.
   This main task is specific to a single-host environment, so it is described under this topic. However, it shares some common tasks with configuring a multi-host environment.
2. “Chapter 5. Configure a new Application Environment” on page 133.
   This main task is common to both single- and multi-host environments, so is described in a later topic.

Each main task includes an example of the complete task, which you can use as a basis for completing and checking the configuration of your own single-host enterprise.

## Plan what to Configure for a Single-host Environment

Before configuring a standalone host, you should identify the components that you need to configure. The decisions that you make when planning your network are generally dictated by the “Name Tree Configuration Management Rules” on page 168. The decisions also influence, and are influenced by, the applications that you want to run on the host.

The easiest way to configure a single-host environment is to use the samples created when you install Component Broker onto the host.

Consider the following points, based on the name tree configuration rules, and the general single-host network shown in Figure 23. From these points, you should be able to create a plan of the following components to configure, like that given in the table Components to Configure for a Single-Host Environment after the figure.

**The Component Broker cell**

The host must be configured as a member of the Component Broker cell. The host serves and prefers that cell. You can choose the name of the cell.

**Workgroups**

The host must be configured as a member of a work group, which is known as the *minimum work group*, required by Component Broker. The host serves and prefers that work group. You can choose the name of the work group.

**Name Servers and TCP/IP Protocols**

The host must have a name server and TCP/IP protocol configured on it. The name that you choose for a name server or TCP/IP protocol is used only to identify the system management object on the System Manager user interface.

**Management Zone and Configuration**

You configure the host within one Configuration of a Management Zone that you should reserve only for configuring your host environment You are recommended to use other Management Zones for your applications. You must activate the Configuration of the Management Zone used for your host environment before any of your application Management Zones.

You must configure the host as serving the cell and work group *before* the Configuration for the host environment is first activated.

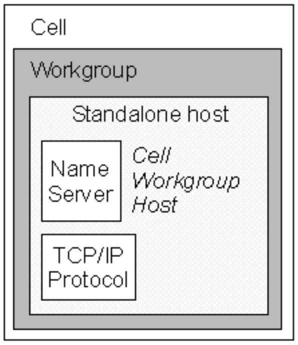


*Figure 23. A Sample Single-Host Environment*

*Table 2.* **Components to Configure for a Single-Host Environment**

| Component | Sample values | Your values? |
|---|---|---|
| **Host** | host1.cbnet.com | |
| **Cell** | Sample Cell | |
| **Work Group** | Sample Work Group | |
| **Name Servers** | Sample Name Server | |
| **TCP/IP Protocols** | Sample TCPIP Protocol | |
| **Management Zone** | Sample Cell and Work Group Zone | |
| **Configuration** | Sample Configuration | |

For information about what you need to configure for an application environment, see “Plan what to Configure for a Single-Host Application Environment” on page 134.

If you want to proceed with configuring a new multi-host network, see “Configure a Single-Host Environment” on page 90.

For an example of configuring a new single-host environment, see “Example: Configure a new Single-host Environment” on page 91.

## Install a Component Broker Single-Host Environment

This topic provides an overview of installing a Component Broker single-host environment, based on the general single-host enterprise shown in Figure 24.

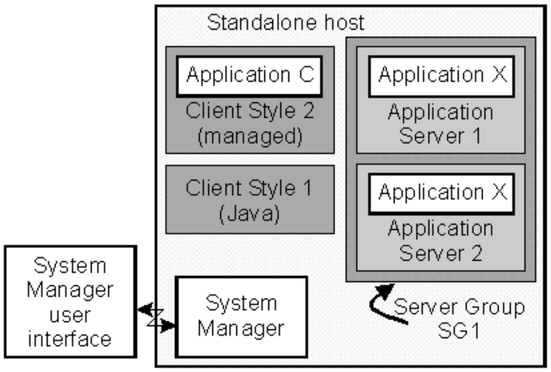


*Figure 24. A Single-Host Enterprise*

The general procedure for creating a Component Broker single-host environment involves the following steps:

1. Install the Component Broker *Standalone Workstation* installation option
2. Complete preliminary system management configuration

After having create your Component Broker host environment, you can configure the host environment then the application environment, as described in the following topics:

- “Configure a Single-Host Environment” on page 90
- “Chapter 5. Configure a new Application Environment” on page 133

For a single-host environment, you use the following Component Broker installation options:

| Installation Option | Description |
|---|---|
| Standalone Workstation | This installs the Component Broker software needed to run the System Manager on the host. It also enables you to run application servers and C++ clients on the host. This host is used to serve the Component Broker cell and minimum work group. |
| Application Client<br>- Java Client<br>- ActiveX Client | Use this option if you want to add ActiveX clients or Java clients to the host. The ActiveX client styles can be administered on the host by the System Manager. (Note that only the C++ and ActiveX client styles can be administered by the System Manager.) You can then use Java clients to access the Component Broker enterprise through this host or another host to which you copy the Java client style properties file. |
| System Manager User Interface | Use this option on another workstation if you want to run the System Manager user interface on a separate workstation. The user interface is used to interact with the System Manager in order to administer the Component Broker enterprise. |

Using the above table, the single-host enterprise diagram can be annotated with the install options needed, as shown in Figure 25.

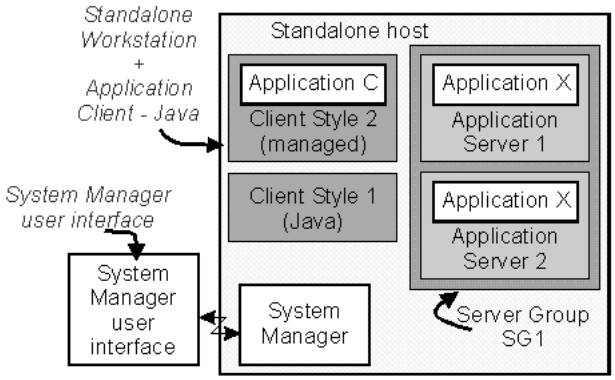


Figure 25. Installation Options for a Single-Host Enterprise

## Install the Component Broker Installation Options

Using the plan of your desired Component Broker enterprise, you can install the Component Broker software (and prerequisite products) onto the standalone host, as described in the *Planning, Performance, and Installation Guide*.

## Complete Preliminary System Management Configuration

Using the plan of your desired Component Broker enterprise, you can use the Component Broker system management configuration tool to perform the preliminary configuration of the standalone host, as described in the *Planning, Performance, and Installation Guide*.

## Integrate System Management with Tivoli

To enable Component Broker system management integration with Tivoli, you need to complete the following tasks:

1. “Install and Configure the Component Broker plus module for Tivoli” on page 47 to add icons to Tivoli, enable its event server to recognize Component Broker events, and define the Tivoli event consoles that are to monitor those events.
2. “configure and enable Tivoli event monitoring” on page 50 so that Component Broker sends events to the Tivoli event server whenever an entry is added to a Component Broker error log.

Also, if you want Tivoli to be able to start the Component Broker System Manager user interface, you must “Install a System Manager User Interface” on page 51 on the same Tivoli host as you installed the Component Broker plus module for Tivoli.

**Related Tasks**

“Install a Component Broker Multi-Host Environment” on page 105.

## Configure a Single-Host Environment

This topic describes how to configure a new standalone Component Broker host into the DCE cell used by Component Broker and the minimum work group required by Component Broker.

The complete task is split into several smaller tasks, to help you configure the host environment more easily in several stages.

An example is provided at the end of this topic.

The task steps assume that you have planned what you want to configure for your host network, as described in “Plan what to Configure for a Single-host Environment” on page 87.

The tasks also assume that you have not already activated a system management Configuration that defines the cell, work group, and serving host of your Component Broker host environment. If you have done this, and want to reorganize your host network, see “Chapter 6. Administer your Host Environment” on page 167.

If you want to use the sample Management Zone provided with Component Broker, follow the steps given in the example, “Example: Use the Sample Configuration for a Single-Host Environment” on page 95 instead of those listed below.

If you need more information about administering your host network, see the related topics listed at the bottom of this page (page 118).

## An Overview of Configuring a Single-Host Environment

The general procedure to configure a new Component Broker single-host environment involves the following tasks:

1. Create a new Management Zone and, within that, a Configuration to organize your host network.

This Management Zone should be reserved only for configuring the host environment into the Component Broker cell and minimum work group. You are recommended to use other Management Zones for your applications. You must activate the Configuration of the Management Zone created in this task before any of your application Management Zones.

2. Configure the Component Broker cell and minimum work group.

   The cell and minimum work group are used to create the name tree needed by Component Broker.

   This task also involves configuring the host as serving both the Component Broker cell and minimum work group, including configuring the name server and protocol for that host. The task ends with activating the Configuration, to check that the Component Broker name tree can be created, and that the Configuration is valid with the one host.

## Configure a Standalone Host

To configure a standalone host, complete the steps given in the following topics. Each topic guides you through to the next topic.

**Note:** These steps are the same as the first steps for configuring a multi-host environment. At a later time, you can add more hosts to your single-host environment to expand the environment into a multi-host network.

1. “Create a new Management Zone and Configuration for your Host Network” on page 110
2. “Configure the Component Broker Cell and Minimum Work Group” on page 111, including the standalone host as serving both the cell and minimum work group

For an example of the complete steps to configure a new single-host environment into the Component Broker cell and minimum work group, see “Example: Configure a new Single-host Environment”.

# Example: Configure a new Single-host Environment

This topic describes how to configure a Component Broker single-host environment, based on the standalone host shown in Figure 26 on page 92. The tasks to be completed configure the host into the DCE cell and minimum work group required by Component Broker. They also configure the name server and TCP/IP protocol needed by the host.

It provides an example of the general procedure described in “Configure a Single-Host Environment” on page 90.

The aim is to guide you through configuring a new single-host Component Broker environment as easily as possible. To achieve this, the task steps are given with only the minimum background information to help you understand the reason for a step. If you need more information, see the related topics listed at the bottom of this page (page 95).

The task steps assume that you have not already activated a system management Configuration that defines the cell, work group, and serving host of the Component Broker host environment. If you have done this, and want to reorganize your host environment, see “Chapter 6. Administer your Host Environment” on page 167.

If you want to use the sample Management Zone provided with Component Broker, follow the steps given in the example, “Example: Use the Sample Configuration for a Single-Host Environment” on page 95 instead of those listed below.

## An Overview of the Example Host Environment

The example host environment (shown in Figure 26) comprises the one standalone host:

- host1.cbnet.com, serves the Component Broker cell *cbcell* and the minimum work group *work group 1*.

The Component Broker System Manager runs on the host. Also, because the single-host environment is intended for testing and as a base for application development, the host is used to run some application servers.

**Note:** In this example host environment, the System Manager user interface and the Java clients run on workstations that are not managed by Component Broker. However, those workstations are configured by Component Broker, to enable the System Manager user interface to communicate with the System Manager and to create appropriate client style properties files on the Java client workstations.

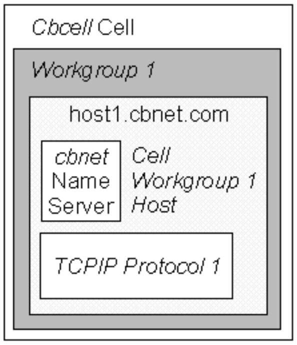


*Figure 26. The Example Host Environment*

## The Task to Configure the Example Single-Host Environment

Use this task to configure the example single-host environment shown in Figure 26. The task creates and configures the system management objects listed in the following table. If you want to use the example to configure your own single-host environment, substitute the names listed with your own names.

*Table 3.* **System Management Objects Configured for the Example Single-host Environment**

| **Object Type** | **Example Name** | **Your Name?** |
|---|---|---|
| Management Zone | Cbroker Network Zone | |
| Configuration | Cbroker Config | |
| Cell | Cbcell | |
| Work Groups | Work group 1 | |
| Hosts | host1.cbnet.com | |
| Name Server | Cbnet | |
| TCP/IP Protocol | TCPIP Protocol 1 | |

**To configure the example single-host environment, complete the following steps:**

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Create the Management Zone for you to configure your host environment.

   This Management Zone should be reserved only for configuring the host into the Component Broker cell and minimum work group. You are recommended to use other Management Zones for your applications. You must activate the Configuration of the Management Zone created in this step before any of your application Management Zones.

   To create a new Management Zone, complete the following steps:

   a. On the pop-up menu of the Management Zones folder, click **Insert**.

      This opens a dialog box for you to specify a unique name for the new Management Zone.

   b. Type *CBroker Network Zone*

   c. To create the new Management Zone, click the **OK** button.

   d. To display the Management Zone, expand the Management Zones folder by clicking its + sign. You should see a Management Zone called *CBroker Network Zone*. If not, repeat the previous steps.
3. Create a Configuration within the Management Zone.

   Each Management Zone must contain at least one Configuration, which defines an implementation of the Management Zone.

   To create a new Configuration, complete the following steps:

   a. On the pop-up menu of the Management Zone *CBroker Network Zone*, click **New - Configuration**.

      This opens a dialog box for you to specify a unique name for the new Configuration.

   b. Type *CBroker Config*

   c. To create the new Configuration, click the **OK** button.

   d. To display the Configuration, expand the Configurations folder of the Management Zone *CBroker Network Zone*. You should see a Configuration called *CBroker Config*. If not, repeat the previous steps.
4. Expand the Configuration *CBroker Config*, by clicking its + sign.

   You create a Cell and the Work Group within the Configuration to configure your host environment. Although you do not need to expand the Configuration at this point, it will help you to see the new folders and objects that you will create in the following steps. When you first create a Configuration, the object folders within it are empty and, by default, not visible.
5. Create the Cell. (The Component Broker host environment requires one, and can contain only one, cell.)

   a. On the pop-up menu of the Configuration *CBroker Config*, click **New - Cell**.

      This displays a dialog box for you to specify a unique name for the Cell.

   b. Type *Cbcell*

   c. To create the Cell, click the **OK** button.

   d. To display the Cell, expand the Cells folder. You should see a Cell called *Cbcell*. If not, repeat the preceding steps.
6. Create the minimum work group. (The Component Broker host environment requires at least one work group, the *minimum work group*.)

a. On the pop-up menu of the Configuration *CBroker Config*, click **New - Work Group**.

   This displays a dialog box for you to specify a unique name for the new Work Group.

b. Type *Work group 1*

c. To create the Work Group, click the **OK** button.

d. To display the Work Group, expand the Work Groups folder. You should see a Work Group called *Work group 1*. If not, repeat the preceding steps.

7. Configure the host, host1.cbnet.com, as serving the cell and minimum work group, and with a TCP/IP Protocol and a Name Server.

   Define and configure the TCP/IP Protocol for the Host:

   a. On the pop-up menu of the Configuration *CBroker Config*, click **New - TCP/IP Protocol**.

      This displays a dialog box for you to specify a unique name for the new Protocol.

   b. Type *TCPIP Protocol 1*

      Note that the name cannot contain a forward slash character (as in *TCP/IP*).

   c. To create the TCP/IP Protocol, click the **OK** button.

   d. To display the Protocol, expand the TCP/IP Protocols folder. You should see an object called *TCPIP Protocol 1*. If not, repeat the preceding steps.

   e. Expand the Hosts folder

   f. On the pop-up menu of the Host *host1.cbnet.com*, click **Drag**.

   g. On the pop-up menu of the TCP/IP Protocol *TCPIP Protocol 1*, click **Configure Host**.

   You can configure the same TCP/IP Protocol onto other Hosts that need a protocol with the same characteristics, as described later.

   Define and configure the Name Server for the Host:

   a. On the pop-up menu of the Configuration *CBroker Config*, click **New - Name Server**.

      This displays a dialog box for you to specify a unique name for the new Name Server.

   b. Type *Cbnet*,

   c. To create the Name Server, click the **OK** button.

   d. To display the Name Server, expand the Name Servers folder. You should see a Name Server called *Cbnet*. If not, repeat the preceding steps.

   e. On the pop-up menu of the Name Server, click **Configure Host**.

      **Note:** The **Configure Host** action was offered because the Host *host1.cbnet.com* is still on the system management clipboard from the last **Drag** action.

8. Define that the host serves the cell and minimum work group. (In any Component Broker host environment there is one host that serves both the cell and the minimum work group.)

   The following steps assume that you have already expanded the Cells folder, and that last object dragged was the Host *host1.cbnet.com* that is to serve the cell. This is true if you followed the preceding steps.

   a. On the pop-up menu of the Cell *Cbcell* click **Configure Serving Host**

   b. On the pop-up menu of the Cell *Cbcell* click **Configure Preferring Host**

c. On the pop-up menu of the Cell *Cbcell* click **Configure Member Host**.

   The following steps assume that you have already expanded the work groups folder, and that last object dragged was the Host *host1.cbnet.com* that is to server the work group. This is true if you followed the preceding steps.

d. On the pop-up menu of the work group *work group 1* click **Configure Serving Host**

e. On the pop-up menu of the work group *work group 1* click **Configure Preferring Host**

f. On the pop-up menu of the work group *work group 1* click **Configure Member Host**.

   **Note:** If a host is to *ever* serve a work group, it must be defined by **Configure Serving Host** before *any* Configuration using the host is first activated.

9. Activate the Configuration. This creates the cell, minimum work group, and name server for the host. It also defines that the host name server is responsible for the runtime management of the cell and work group name tree.

   On the pop-up menu of the Configuration *CBroker Config*, click **Activate**.

   The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

You have now configured the single-host environment shown in Figure 26 on page 92.

The next stage is to create the application environment that is to use the host environment, as described in “Example: Configure a new Single-host Application Environment” on page 154. This adds the servers, client styles, and applications that are to run on the host.

## Example: Use the Sample Configuration for a Single-Host Environment

This topic describes how to use the samples provided with Component Broker to configure a single-host environment, based on the standalone host shown in the figure The Sample Host Environment (page Figure 27 on page 96). The tasks to be completed configure the host into the DCE cell and minimum work group required by Component Broker. They also configure the name server and TCP/IP protocol needed by the host.

It provides an example of the general procedure described in “Configure a Single-Host Environment” on page 90.

Using the samples provided with Component Broker is the easiest way to configure your host environment. If you want to use the sample names, all you need to do is activate the Sample Configuration. Otherwise, you can change the sample names to your own names, then just activate the Configuration.

The task steps assume that you have not already activated a system management Configuration that defines the cell, work group, and serving host of the Component Broker host environment. If you have done this, and want to reorganize your host environment, see “Chapter 6. Administer your Host Environment” on page 167.

## An Overview of the Sample Host Environment

The sample host environment (shown in Figure 27) comprises the one standalone host:

- host1.cbnet.com, serves the Component Broker cell *Sample Cell* and the minimum work group *Sample Work Group*.

The Component Broker System Manager runs on the host. Also, because the single-host environment is intended for testing and as a base for application development, the host is used to run some application servers.

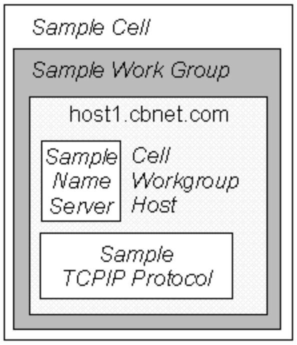


*Figure 27. The Sample Host Environment*

The following figure shows how the sample host environment is represented through the System Manager user interface.

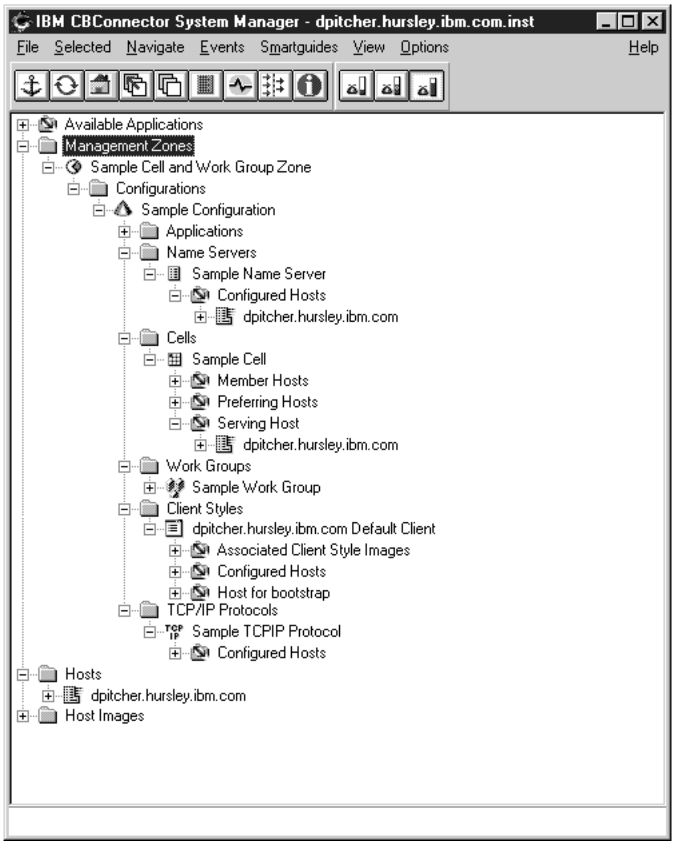


*Figure 28. The Sample Host Environment, as shown by the System Manager user interface*

## The Task to Configure the Sample Single-Host Environment

Use this task to configure the sample single-host environment shown in Figure 27 on page 96. The task creates and configures the system management objects listed in the following table. If you want to use the example to configure your own single-host environment, substitute the names listed with your own names.

*Table 4.* **System Management Objects Configured for the Sample Single-host Environment**

| Object Type | Sample Name | Your Name? |
|---|---|---|
| Management Zone | Sample Cell and Work Group Zone | |
| Configuration | Sample Configuration | |
| Cell | Sample Cell | |
| Work Groups | Sample Work Group | |
| Hosts | host1.cbnet.com | |
| Name Server | Sample Name Server | |
| TCP/IP Protocol | Sample TCPIP Protocol | |

Using the samples provided with Component Broker is the easiest way to configure your host environment, as described in either of the following topics:

- If you want to use the sample names, all you need to do is activate the Sample Configuration. To continue, see "Configure the Sample Single-Host Environment Using the Sample Names".
- Otherwise, you can change the sample names to your own names, then just activate the Configuration. To continue, see "Configure the Sample Single-Host Environment Using your own Names" on page 99.

**Note:** Besides the system management obects listed above, the Sample Configuration contains a default client style (called *hostname* Default Client Style) for the host and some default applications for Component Broker services. You should not rename those system management objects.

## Configure the Sample Single-Host Environment Using the Sample Names

To configure the sample single-host environment using the sample names, complete the following steps:

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Expand the Management Zones folder
3. Expand *Sample Cell and Work Group Zone*
4. Expand the Configurations folder
5. Activate the Sample Configuration. This creates the cell, minimum work group, and name server for the host. It also defines that the host name server is responsible for the runtime management of the cell and work group name tree.

   On the pop-up menu of *Sample Configuration*, click **Activate**.

   The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have installed Component Broker properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, you should see an *Activation successful* message.

If you see an error message in the Action Console window about the cell already being created, a Configuration that defines the Component Broker cell has already been activated. To continue, you can either remove the other Configuration then activate the sample Configuration again or reconfigure your host environment using the sample Configuration, as described in "Chapter 6. Administer your Host Environment" on page 167.

**Check that the Component Broker name tree has been created successfully**

Optionally, to verify that the host name server is properly stored in the DCE Director Cell Directory Service (CDS), use the following task "Verify that a Name Server is in the DCE CDS" on page 119.

For a standalone host this finishes the configuration of the single-host environment. Next, you can configure the application environment that is to run on that host, as described in Configure the Sample Application Environment. This adds the servers, client styles, and applications that are to run on the host.

## Configure the Sample Single-Host Environment Using your own Names

To configure a single-host environment, but change the sample names, complete the following steps:

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Expand the Management Zones folder
3. Expand *Sample Cell and Work Group Zone*. To rename the Management Zone, see “To rename a system management object” on page 100.
4. Expand the Configurations folder
5. Expand *Sample Configuration*. To rename the Configuration, see “To rename a system management object” on page 100.
6. To display the sample Name Server, expand the Name Servers folder. To rename the Name Server, see “To rename a system management object” on page 100.
7. To display the sample Cell, expand the Cells folder. To rename the Cell, see “To rename a system management object” on page 100.
8. To display the sample Work Group, expand the Work Groups folder. To rename the Work Group, see “To rename a system management object” on page 100.
9. To display the sample TCP/IP Protocol, expand the TCP/IP Protocols folder. To rename the TCP/IP Protocol, see “To rename a system management object” on page 100.
10. Activate the Configuration. This creates the cell, minimum work group, and name server for the host. It also defines that the host name server is responsible for the runtime management of the cell and work group name tree.

    On the pop-up menu of *Sample Configuration*, click **Activate**

    The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have installed Component Broker properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, you should see an *Activation successful* message.

If you see an error message in the Action Console window about the cell already being created, a Configuration that defines the Component Broker cell has already been activated. To continue, you can either remove the other Configuration then activate the sample Configuration again or reconfigure your host environment using the sample Configuration, as described in “Chapter 6. Administer your Host Environment” on page 167.

**Check that the Component Broker name tree has been created successfully**

Optionally, to verify that the host name server is properly stored in the DCE Director Cell Directory Service (CDS), use the following task “Verify that a Name Server is in the DCE CDS” on page 119.

For a standalone host this finishes the configuration of the single-host environment. Next, you can configure the application environment that is to run on that host, as described in Configure the Sample Application Environment. This adds the servers, client styles, and applications that are to run on the host.

## To rename a system management object

To change the name of any system management object, you use the following procedure:

1. On the pop-up menu of the object, click **Rename**
2. In the Object Rename dialog window, type the new name for the object
3. To apply the change, click the **OK** button.

# Chapter 4. Create a New Multi-Host Enterprise

This topic describes how to create and configure a new Enterprise that uses several managed hosts to support applications that are to run on several application servers and client styles.

The tasks used are based on the following assumptions:

- You have already installed Component Broker onto the hosts to be configured, as described in the *Planning, Performance, and Installation Guide*. For an overview of installing a Component Broker multi-host environment, see “Install a Component Broker Multi-Host Environment” on page 105.
- You have not already activated a system management Configuration that defines the cell, work group, and serving hosts of your Component Broker host environment. If you have done this, and want to reorganize your host environment, see “Chapter 6. Administer your Host Environment” on page 167.

If you want to use the sample Management Zone provided with Component Broker, follow the steps given in the example, “Example: Use the Sample Configuration for a Multi-Host Environment” on page 126 instead of those listed below.

If you are already familiar with using the System Manager user interface, and have a good plan of what you want your enterprise to be, you can skip the initial overview, and start by “Configure a Multi-Host Environment” on page 109.

## Overview of Configuring a Multi-host Enterprise

This topic describes things that you should consider when preparing to create a new multi-host enterprise, based on the general multi-host enterprise shown in Figure 29 on page 102.

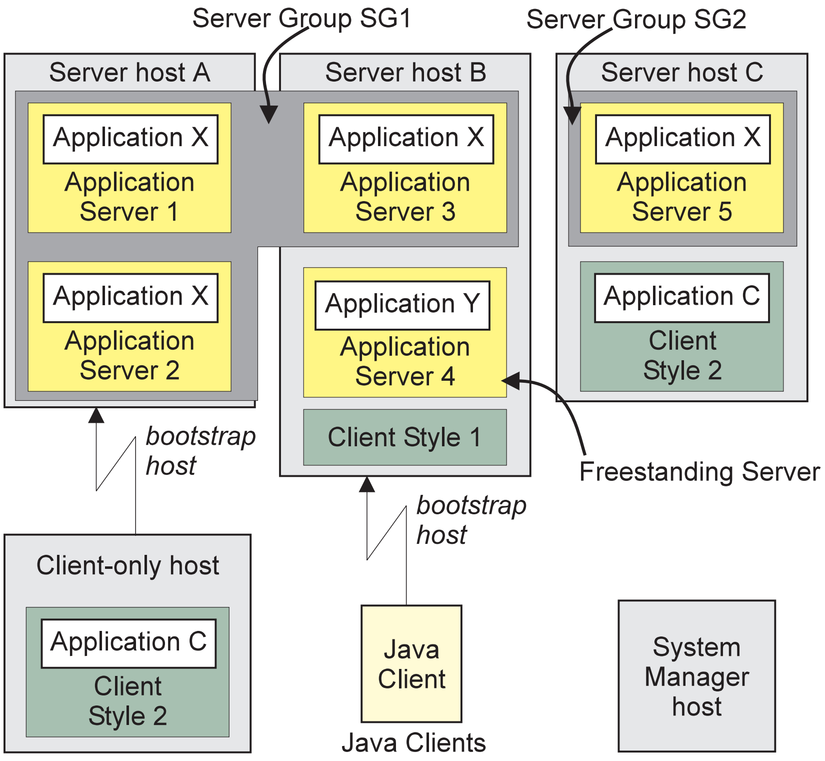


*Figure 29.* **A Multi-Host Enterprise**

When preparing to create any enterprise, the first aim is to get a clear plan of what you want to create. The planning and creation of any enterprise can be simplified by considering the enterprise in two parts; the *host environment* and the *application environment*.

**The Host Environment**

The complete Component Broker host environment comprises the server hosts on which your applications run, managed client-only hosts, and perhaps a separate System Manager host.

When creating a new enterprise, you need to configure the environment of managed server hosts into the Component Broker cell, the minimum work group, and any other work groups that you want to use. Also, each server host in the host environment needs to be configured with a name server and TCP/IP protocol.

For more information about what you need to configure for a multi-host environment, see “Plan what to Configure for a Multi-host Network” on page 103.

**The Application Environment**

The application environment comprises the clients, servers, and applications that run on the your Component Broker host environment. It also can include clients that run on workstations and other hosts that are not managed by Component Broker; for example, Java clients.

When creating a new enterprise, you need to configure the application environment into server groups, freestanding servers, and client styles. You also need to load applications into Component Broker and configure those applications as instructed by information provided with each application.

For more information about what you need to configure for an application environment, see “Plan what to Configure for a Multi-Host Application Environment” on page 137.

If you want to proceed with configuring a new multi-host enterprise, complete the following main tasks:

1. “Configure a Multi-Host Environment” on page 109
   This main task is specific to a multi-host environment, so it is described under this topic. However, it shares some common tasks with configuring a single-host environment.
2. “Chapter 5. Configure a new Application Environment” on page 133.
   This main task is common to both single- and multi-host environments, so is described in a later topic.

Each main task includes an example of the complete task, which you can use as a basis for completing and checking the configuration of your own multi-host enterprise.

## Plan what to Configure for a Multi-host Network

Before configuring your multi-host network, you should create a plan of the network and identify the components that you need to configure. The decisions that you make when planning your network are generally dictated by the “Name Tree Configuration Management Rules” on page 168. The decisions also influence, and are influenced by, the applications that you want in your enterprise and how you want to distribute those applications.

Consider the following points, based on the name tree configuration rules, and the general multi-host network shown in Figure 30 on page 104. From these points, you should be able to create a plan of the following components to configure, like that given in the table Components to Configure for a Multi-Host Network (page Figure 23 on page 87) after the figure.

**The Component Broker cell**

The System Manager host serves the cell; that is, that host provides the name server for the cell. All server hosts managed by the same System Manager must prefer, and be members of, the one cell. You can choose the name of the cell.

**Workgroups**

The server hosts in your network can be grouped into one or more work groups.

- You can choose the name of each work group.
- The System Manager host also serves a work group, which is known as the *minimum work group* required for a Component Broker network. At least one host must be a member of that work group.
- To configure another work group, one host must serve the work group and at least one host must be a member of the work group.
- If a host is to *ever* serve any work group, it must be configured as serving a work group when it is first configured into the network. So you must be sure of whether or not a host is to serve a work group before you first configure the host.

- Each server host must prefer one work group, and must be a member of that preferred work group. The work group that a host prefers cannot be changed, so you must be sure of a host's preferred work group before you first configure the host.
- Each host can also be a member of other work groups.

**Name Servers and TCP/IP Protocols**

The System Manager host and all managed server hosts must have a name server and TCP/IP protocol configured on them. The name that you choose for a name server or TCP/IP protocol is used only to identify the system management object on the System Manager user interface. All the hosts can use the same Name Server and TCP/IP Protocol definition if they require the same characteristics. You can also define other Name Servers and TCP/IP Protocols for hosts that require different characteristics.

Managed client-only hosts must have a TCP/IP protocol configured on them, but do not have a name server.

**Management Zone and Configuration**

You configure your host network within one Configuration of a Management Zone that you should reserve only for configuring your host network. You are recommended to use other Management Zones for your applications. You must activate the Configuration of the Management Zone used for your network before any of your application Management Zones.

Managed client-only hosts are configured as part of the application environment.

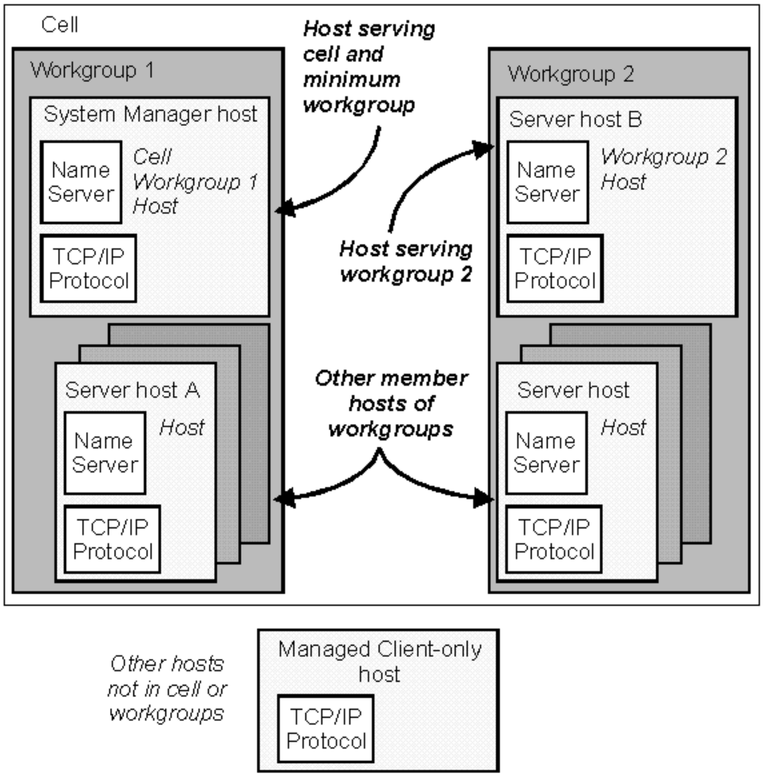


*Figure 30. A General Multi-Host Network*

*Table 5.* **Components to Configure for a Multi-Host Network**

| **Hosts** | **System Manager** | **A** | **B** | **C** | **Managed client-only** | **...** | |
|---|---|---|---|---|---|---|---|
| **Cell cell_name** | Serve | Prefer | Prefer | Prefer | n/a | ... | |
| **Workgroups** | | | | | | | |
| work group_1 | Serve, Prefer | Member | Member | Prefer | n/a | ... | |
| work group_2 | - | Member | Serve, Prefer | - | n/a | ... | |
| **Name Servers** | | | | | | | |
| server_name | uses | uses | uses | uses | n/a | ... | |
| **TCP/IP Protocols** | | | | | | | |
| protocol_name | uses | uses | uses | uses | uses | ... | |
| **Management Zone** | zone_name | | | | | | |
| **Configuration** | config_name | | | | | | |
| **Notes:** | *Prefer* means the host prefers the cell or work group, and so is a member of the cell or work group.<br>*n/a* means the option is not applicable (for managed client-only hosts) | | | | | | |

For information about what you need to configure for an application environment, see “Plan what to Configure for a Multi-Host Application Environment” on page 137.

If you want to proceed with configuring a new multi-host network, see “Configure a Multi-Host Environment” on page 109.

For an example of configuring a new multi-host network, see “Example: Configure a new Multi-Host Network” on page 121.

## Install a Component Broker Multi-Host Environment

This topic provides an overview of installing a Component Broker multi-host environment, based on the general multi-host enterprise shown in Figure 31 on page 106.

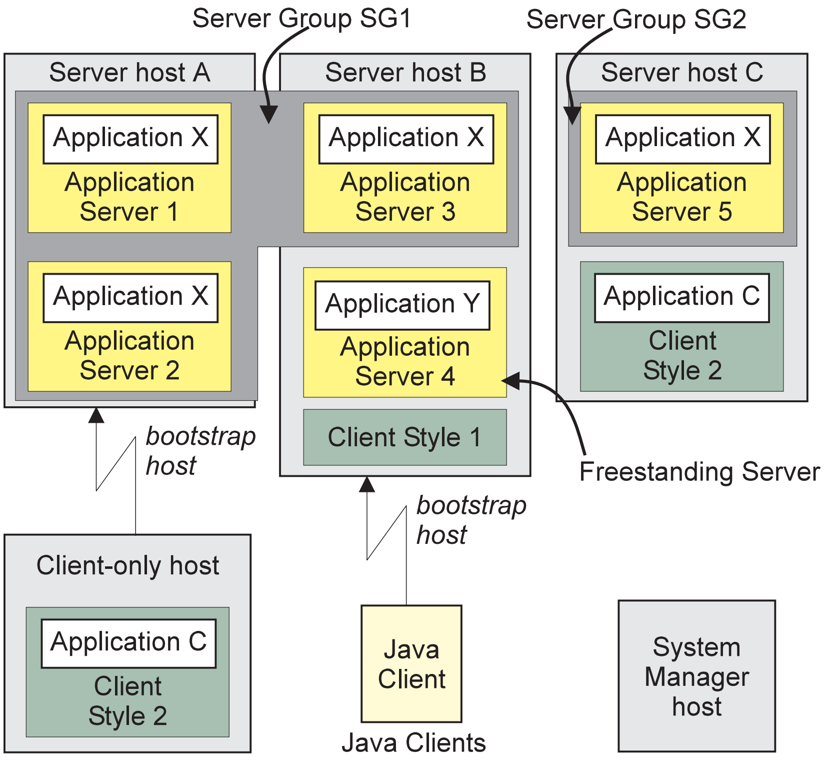


*Figure 31.* **A Multi-Host Enterprise**

The general procedure for creating a Component Broker host environment involves the steps:

1. Prepare a plan of the host environment, identifying the types of host and therefore the Component Broker installation options needed
2. For each host:
   a. Install the Component Broker installation options
   b. Complete preliminary system management configuration

After having created your Component Broker host environment, you can configure the host environment then the application environment, as described in the following topics:

- “Configure a Multi-Host Environment” on page 109
- “Chapter 5. Configure a new Application Environment” on page 133

## Plan the Host Environment

Before installing your multi-host environment, you should create a plan of the environment, identifying the types of host and therefore the Component Broker installation options needed.

The decisions that you make when planning your host environment are generally dictated by the applications that you want in your enterprise and how you want to distribute those applications.

You can identify the following types of hosts, and the associated Component Broker installation options:

| Host Type | Installation Option | Description |
|---|---|---|
| System Manager host | System Manager | This installs the Component Broker software needed to run the System Manager on the host. The System Manager is used to administer all the managed server hosts and managed client-only hosts in the host environment. This host is used to serve the Component Broker cell and minimum work group. |
| Managed Server Host | Application Server | Use this option if you want to run application servers on the host. It also enables you to run C++ clients on the host. |
| Managed Client-only Host | Application Client<br>- C++ Client<br>- ActiveX Client | Use this option if you want to run C++ clients or ActiveX clients on the host. These client styles can be administered on the host by the System Manager. Note that only the C++ and ActiveX client styles can be administered by the System Manager. |
| Java Client Host | Application Client<br>- Java Client | Use this option if you want to configure Java client style properties files on the host. You can then use Java clients to access the Component Broker enterprise through this host or another host to which you copy the Java client style properties file. |
| System Manager user interface host | System Manager User Interface | Use this option on a managed host or another workstation where you want to run the System Manager user interface. The user interface is used to interact with the System Manager in order to administer the Component Broker enterprise. |

Using the above table, the multi-host enterprise diagram can be annotated with the install options needed, as shown in Figure 32 on page 108.

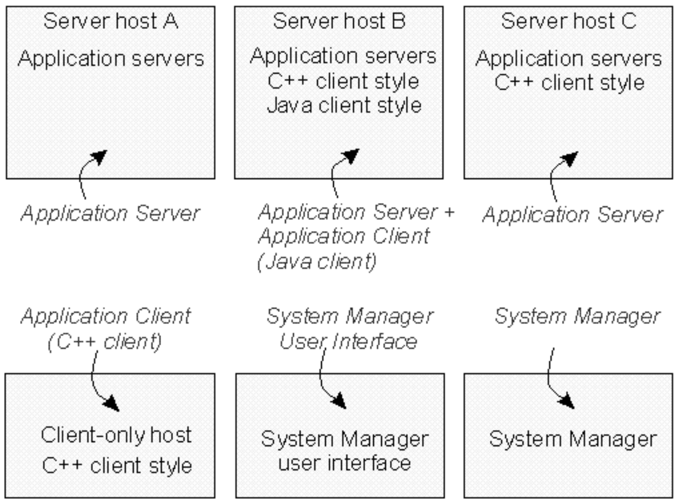


*Figure 32.* **Installation Options for a Multi-Host Enterprise**

## Install the Component Broker Installation Options

Using the plan of your desired Component Broker enterprise, you can install the Component Broker software (and prerequisite products) onto each host, as described in the *Planning, Performance, and Installation Guide*.

## Complete Preliminary System Management Configuration

Using the plan of your desired Component Broker enterprise, you can use the Component Broker system management configuration tool to perform the preliminary configuration of each host, as described in the *Planning, Performance, and Installation Guide*.

## Integrate System Management with Tivoli

To enable Component Broker system management integration with Tivoli, you need to complete the following tasks:

1. “Install and Configure the Component Broker plus module for Tivoli” on page 47 to add icons to Tivoli, enable its event server to recognize Component Broker events, and define the Tivoli event consoles that are to monitor those events.
2. “configure and enable Tivoli event monitoring” on page 50 so that Component Broker sends events to the Tivoli event server whenever an entry is added to a Component Broker error log.

Also, if you want Tivoli to be able to start the Component Broker System Manager user interface, you must “Install a System Manager User Interface” on page 51 on the same Tivoli host as you installed the Component Broker plus module for Tivoli.

**Related Tasks**

“Chapter 4. Create a New Multi-Host Enterprise” on page 101

# Configure a Multi-Host Environment

This topic describes how to configure a new Component Broker multi-host environment so that the hosts are organized into the DCE cell used by Component Broker and into one or more workgroups.

The complete task is split into several smaller tasks, to help you develop your environment more easily in several stages. Some of these tasks can also be used at a later time to add hosts and workgroups to your environment.

An example is provided at the end of this topic.

The task steps assume that you have planned what you want to configure for your host environment, as described in "Plan what to Configure for a Multi-host Network" on page 103.

The tasks also assume that you have not already activated a system management Configuration that defines the cell, work group, and serving hosts of your Component Broker host environment. If you have done this, and want to reorganize your host environment, see "Chapter 6. Administer your Host Environment" on page 167.

If you want to use the sample Management Zone provided with Component Broker, follow the steps given in the example, "Example: Use the Sample Configuration for a Multi-Host Environment" on page 126 instead of those listed below.

If you need more information about administering your host environment, see the related topics listed at the bottom of this page (page 118).

## An Overview of Configuring a Multi-host Environment

The general procedure to configure a new Component Broker multi-host environment involves the following tasks:

1. Create a new Management Zone and, within that, a Configuration to organize your host environment.

   This Management Zone should be reserved only for configuring your host environment into the Component Broker cell and one or more workgroups. You are recommended to use other Management Zones for your applications. You must activate the Configuration of the Management Zone created in this task before any of your application Management Zones.

2. Configure the Component Broker cell and minimum work group.

   The cell and minimum work group are used to create the name tree needed by Component Broker.

   This task also involves configuring the System Manager host, which serves both the Component Broker cell and minimum work group, including configuring the name server and protocol for that host. The task ends with activating the Configuration, to check that the Component Broker name tree can be created, and that the Configuration is valid with the one host.

3. If required, define another work group and configure the host that serves that work group.

   This task also involves configuring the host that serves the work group, including configuring the name server and protocol for that host. The task ends with activating the Configuration, to check that the Configuration is valid and to add the new work group and host to the name tree.

4. Configure other hosts that are members of the workgroups.

   This task includes configuring the name server and protocol for each host, and configuring the work group that each host prefers. The task ends with activating the Configuration, to check that the Configuration is valid and to add the new hosts to the name tree.

## Configure a Multi-Host Environment

To configure your multi-host environment, complete the steps given in the following topics. Each topic guides you through to the next topic.

1. “Create a new Management Zone and Configuration for your Host Network”
2. “Configure the Component Broker Cell and Minimum Work Group” on page 111, including the System Manager host that serves both the cell and minimum work group
3. If required, “Configure a new Work Group” on page 115, including the host that serves the work group
4. “Configure a new Host into your Network” on page 117.

For an example of the complete steps to configure a new multi-host environment into the Component Broker cell and two workgroups, see “Example: Configure a new Multi-Host Network” on page 121.

## Create a new Management Zone and Configuration for your Host Network

Complete the steps in this topic to create a new Management Zone and, within that, a Configuration to organize your host network.

This Management Zone should be reserved only for configuring your host environment (*all the hosts managed by the same System Manager*) into a cell and one or more workgroups. You are recommended to use other Management Zones for your applications. You must activate the Configuration of the Management Zone created in this task before any of your application Management Zones.

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. To create a new Management Zone, complete the following steps:
   a. On the pop-up menu of the Management Zones folder, click **Insert**

      This opens a dialog box for you to specify a unique name for the new Management Zone. Type the name that you want the new object to be known by. The name can contain:

      - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
      - Embedded blanks

   b. To create the new Management Zone, click the **OK** button. If you specified a valid name, this creates a new Management Zone. If the name specified is not valid, a dialog box is displayed for you to enter a new name.

3. Expand the Management Zones folder, by clicking its + sign. You should see an object for the Management Zone that you created. If not, repeat the previous step.
4. Create a Configuration within your Management Zone.

   Each Management Zone must contain at least one Configuration, which defines an implementation of the Management Zone.

   a. On the pop-up menu of your Management Zone, click **New - Configuration**.

      This opens a dialog box for you to specify a unique name for the new Configuration. Type the name that you want the new object to be known by. The name can contain:

      - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
      - Embedded blanks

   b. To create the new Configuration, click the **OK** button. If you specified a valid name, this creates a new Configuration. If the name specified is not valid, you are prompted to enter a new name.

   c. Expand the Configurations folder of your Management Zone. You should see an object for the Configuration that you created. If not, repeat the previous steps.

Next “Configure the Component Broker Cell and Minimum Work Group”, including the host that serves both the cell and minimum work group.

## Configure the Component Broker Cell and Minimum Work Group

Use this task to configure the cell and minimum work group used to create the Component Broker name tree.

This task also involves configuring the System Manager host, which serves both the Component Broker cell and minimum work group, including configuring the name server and protocol for that host. The task ends with activating the Configuration, to check that the Component Broker name tree can be created, and that the Configuration is valid with the one host.

**Note:** In a single-host environment, that host is the System Manager host.

The System Manager host (which serves the cell) must also serve a work group, referred to as the *minimum work group*.

In a multi-host network, all managed server hosts in your Component Broker network must be members of, and prefer, the one cell. You can group your server hosts into the minimum work group and any other workgroups that you create, as described in later tasks.

**Prerequisites:**

This topic assumes that you have created a Management Zone and Configuration to be used to configure your Component Broker network. You create the Management Zone and its initial Configuration, as described in “Create a new Management Zone and Configuration for your Host Network” on page 110. You can also add alternative Configurations to that Management Zone.

**To configure the Component Broker cell and minimum work group, complete the following steps:**

1. Expand the Configuration of the Management Zone that you want to use to organize your host environment.

   You create a Cell and one or more Workgroups within the Configuration. Although you do not need to expand the Configuration at this point, it will help you to see the new folders and objects that you will create in the following steps. When you first create a Configuration, the object folders within it are empty and, by default, not visible.
2. Create the Cell.
   a. On the pop-up menu of the Configuration, click **New - Cell**.

      This displays a dialog box for you to specify a unique name for the Cell.

   b. Type the name that you want the Cell to be known by. The name can contain:
      - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
      - Embedded blanks

      For example, type *cbcell*.

   c. To create the Cell, click the **OK** button. If the name is valid, this creates an object in the Cells folder. If the name is not valid, you are prompted to enter a new name.

   d. To display the Cell, expand the Cells folder. You should see a Cell object with the name that you just specified. If not, repeat the preceding steps.
3. Create the minimum work group.
   a. On the pop-up menu of the Configuration, click **New - Work Group**.

      This displays a dialog box for you to specify a unique name for the new Work Group.

   b. Type the name that you want the Work Group to be known by. The name can contain:
      - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
      - Embedded blanks

      For example, type *Work Group 1*.

   c. To create the Work Group, click the **OK** button. If the name is valid, this creates an object in the Workgroups folder. If the name is not valid, you are prompted to enter a new name.

   d. To display the Work Group, expand the Workgroups folder. You should see a Work Group object with the name that you just specified. If not, repeat the preceding steps.
4. Configure the System Manager host (to serve the cell and minimum work group) with a TCP/IP Protocol and a Name Server.

   **Define the TCP/IP Protocol for the Host.**

   If your Configuration already contains a TCP/IP Protocol that you want to use, you should continue to Configure the TCP/IP Protocol onto the Host (page 113). Otherwise, first define a new TCP/IP Protocol, as described in the following steps:

   a. On the pop-up menu of your Configuration, click **New - TCP/IP Protocol**.

This displays a dialog box for you to specify a unique name for the new Protocol.

b. Type the name that you want the Protocol to be known by. The name can contain:
   - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks

   This name is used only to identify the protocol on the System Manager user interface.

c. To create the object, click the **OK** button. If the name is valid, this creates an object in the TCP/IP Protocols folder. If the name is not valid, you are prompted to enter a new value.

d. To display the Protocol, expand the TCP/IP Protocols folder. You should see an object with the name that you just specified. If not, repeat the preceding steps.

**Configure the TCP/IP Protocol onto the Host model that defines the System Manager host:**

a. Expand the Hosts folder

b. On the pop-up menu of the Host that defines the System Manager host, click **Drag**.

c. On the pop-up menu of the TCP/IP Protocol, click **Configure Host**.

You can configure the same TCP/IP Protocol onto other Hosts that need a protocol with the same characteristics, as described in other tasks.

**Define the Name Server for the System Manager host.**

If your Configuration already contains a Name Server that you want to use, you should continue to Configure the Name Server onto the Host (page 113).

a. On the pop-up menu of your Configuration, click **New - Name Server**.

   This displays a dialog box for you to specify a unique name for the new Name Server.

b. Type the name that you want the Name Server to be known by. The name can contain:
   - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks

c. To create the Name Server, click the **OK** button. If the name is valid, this creates an object in the Name Servers folder. If the name is not valid, you are prompted to enter a new name.

d. To display the Name Server, expand the Name Servers folder. You should see a Name Server with the name that you just specified. If not, repeat the preceding steps.

**Configure the Name Server for the System Manager host:** On the pop-up menu of the Name Server, click **Configure Host**.

**Note:** The **Configure Host** action was offered because the Host is still on the system management clipboard from the last **Drag** action.

You can configure the same Name Server onto other Hosts that need a protocol with the same characteristics, as described in other tasks.

5. By default, the name server reads data from the most convenient DCE/CDS server. However, you can configure the name server to get data from the CDS clerk's cache or from the server where the master replica is located, depending on the degree to which the name server can trust the accuracy of data returned from the DCE/CDS server. You do this by "Change the DCE/CDS Confidence Level" on page 175.

6. Define the System Manager host as serving the cell and minimum work group.

   The following steps assume that you have already expanded the Cells folder, and that the last object dragged was the Host that defines the System Manager host. This is true if you followed the preceding steps.

   a. On the pop-up menu of the Cell click **Configure Serving Host**

   b. On the pop-up menu of the Cell click **Configure Preferring Host**

      If a pop-up window is displayed with a cardinality error message, you have already activated a Configuration that defines the Component Broker cell. For example, you may have used the same System Manager to activate the sample Configuration described in the *Quick Beginnings*.

      To continue, it is perhaps easiest to continue using the existing Configuration that has already been activated. (You can rename that Configuration, and adapt it to your new requirements; for example, as described in "Example: Use the Sample Configuration for a Multi-Host Environment" on page 126.) If you want to continue with a new Configuration, you first need to remove the Name Servers on each of the hosts used by the existing Configuration. (See "Remove a Name Server from a Host" on page 176.) Only then can you complete the configuration of the cell and be able to activate the new Configuration that defines the cell.

   c. On the pop-up menu of the Cell click **Configure Member Host**.

      The following steps assume that you have already expanded the Workgroups folder, and that last object dragged was the Host that defines the System Manager host. This is true if you followed the preceding steps.

   d. On the pop-up menu of the Work Group click **Configure Serving Host**

   e. On the pop-up menu of the Work Group click **Configure Preferring Host**

   f. On the pop-up menu of the Work Group click **Configure Member Host**.

      **Note:** If a host is to *ever* serve a work group, it must be defined by **Configure Serving Host** before *any* Configuration using the host is first activated.

7. Activate the Configuration. This creates the cell, minimum work group, and name server for the System Manager host. It also configures that host with the TCP/IP protocol and defines that the host name server is responsible for the runtime management of the cell and work group name tree.

   On the pop-up menu of the Configuration, click **Activate**.

   The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

If a pop-up window is displayed with a cardinality error message, you have already activated a Configuration that defines the Component Broker cell. For example, you may have used the same System Manager to activate the sample Configuration described in the *Quick Beginnings*. To continue, it is perhaps easiest to continue using the existing Configuration that has already been activated. (You can rename that Configuration, and adapt it to your new requirements; for example, as described in “Example: Use the Sample Configuration for a Multi-Host Environment” on page 126.) If you want to continue with a new Configuration, you first need to remove the Name Servers on each of the hosts used by the existing Configuration. (See “Remove a Name Server from a Host” on page 176.) Only then can you complete the configuration of the cell and be able to activate the new Configuration that defines the cell.

**Check that the Component Broker name tree has been created successfully**

Optionally, to verify that the host name server is properly stored in the DCE Director Cell Directory Service (CDS), use the following task “Verify that a Name Server is in the DCE CDS” on page 119.

For a standalone host this finishes the configuration of the single-host environment. Next, you can configure the application environment that is to run on that host, as described in “Chapter 5. Configure a new Application Environment” on page 133.

Next, if required for a multi-host network, “Configure a new Work Group”, including the host that serves the work group.

Otherwise, for a multi-host network, “Configure a new Host into your Network” on page 117 .

## Configure a new Work Group

Use this task to define a new work group, extra to the minimum work group required by Component Broker. This task also involves configuring the host that serves the work group and, if required, configuring the name server and protocol for that host. The task ends with activating the Configuration, to check that the Configuration is valid and to add the new work group and host to the name tree.

**Note:** You are advised to configure the System Manager host, which serves the Component Broker cell and minimum work group, as serving the new work group.

**Prerequisites:**

This task makes the following assumptions:

- You have already installed Component Broker onto the host that serves the work group. If not, see the *Planning, Performance, and Installation Guide* for information.
- You have already configured the Component Broker cell and minimum work group. Otherwise, see “Configure the Component Broker Cell and Minimum Work Group” on page 111.
- You have already defined the TCP/IP Protocol that the host is to use. You can use an existing protocol, such as the one used when you configured the Component Broker cell and minimum work group. Otherwise, see “Define a TCP/IP or IPC Protocol” on page 170.

- You have already defined the Name Server that the host is to use. You can use an existing Name Server, such as the one used when you configured the Component Broker cell and minimum work group. Otherwise, see “Define a Name Server” on page 171.

**To configure a new work group, and configure the host that serves the work group, complete the following steps:**

1. Expand the Configuration of the Management Zone that you are using to configure your host network.
2. Define the work group.
   a. On the pop-up menu of the Configuration, click **New - Work Group**.

      This displays a dialog box for you to specify a unique name for the new Work Group.

   b. Type the name that you want the Work Group to be known by. The name can contain:
      - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
      - Embedded blanks

      For example, type *Work Group 2*.

   c. To create the Work Group, click the **OK** button. If the name is valid, this creates an object in the Work Groups folder. If the name is not valid, you are prompted to enter a new name.
   d. To display the Work Group, expand the Work Groups folder. You should see a Work Group object with the name that you just specified.
3. Configure the TCP/IP Protocol for the Host that serves the work group:
   a. Expand the Hosts folder
   b. On the pop-up menu of the Host, click **Drag**
   c. Expand the TCP/IP Protocols folder
   d. On the pop-up menu of the TCP/IP Protocol, click **Configure Host**
4. Configure the name server for the host:
   a. Expand the Name Servers folder
   b. On the pop-up menu of the Name Server, click **Configure Host**

      **Note:** The **Configure Host** action was offered because the Host is still on the system management clipboard from the last **Drag** action.
5. If the host is not already configured as a member of the Component Broker cell, configure the host as a member of, and preferring, that cell:
   a. On the pop-up menu of the Cell, click **Configure Preferring Host**
   b. On the pop-up menu of the Cell, click **Configure Member Host**.

      All hosts in your Component Broker network must be members of, and prefer, the one cell.
6. Configure the host as serving the new work group.

   (If a host is to ever serve any work group, it must be configured as serving a work group before any Configuration that uses that host is activated. So the host must be either a new host that you are adding to your network or already have a *Served Work Groups* relationship.)

   On the pop-up menu of the Work Group, click **Configure Serving Host**.
7. If the host is not already configured as preferring one of the work groups that it serves, configure the preferred work group:

   a. On the pop-up menu of the preferred Work Group, click **Configure Preferring Host**
   b. On the pop-up menu of the preferred Work Group, click **Configure Member Host**.
8. If the host is to be a member of other existing work groups, configure the host as a member of those workgroups. Otherwise, if you have not yet defined a work group that you need to act on, you can add the host as a member later.

   On the pop-up menu of each Work Group, click **Configure Member Host**.

Repeat the steps in this topic for each work group that you want to use for your host network.

Optionally, to verify that the host name server is properly stored in the DCE Director Cell Directory Service (CDS), use the following task "Verify that a Name Server is in the DCE CDS" on page 119.

Next, "Configure a Host as a Member of Non-preferred Work Groups" on page 118.

**Related Concepts**

"System Management Representation of Your Enterprise" on page 37
"The Model World" on page 39

**Related Tasks**

"Configure the Component Broker Cell and Minimum Work Group" on page 111
"Configure a new Host into your Network"
"Configure a Host as a Member of Non-preferred Work Groups" on page 118

## Configure a new Host into your Network

Use this task to configure a new host into your Component Broker network.

This task includes configuring the name server and protocol for the host, configuring the host into the Component Broker cell, and configuring the work group that the host prefers. The task ends with activating the Configuration, to check that the Configuration is valid and to add the new hosts to the name tree.

**Note:** All hosts are configured as a member of, and preferring, the Component Broker cell. A host can be a member of one or more work groups, but can prefer only one work group (of which it must be a member).

**Prerequisites:**

This task makes the following assumptions:

- You have already installed Component Broker onto the host. If not, see the *Planning, Performance, and Installation Guide* for information.
- You have already configured the Component Broker cell and minimum work group. Otherwise, see "Configure the Component Broker Cell and Minimum Work Group" on page 111.
- If the host prefers a work group that is served by another host, or is a member of another work group, you have already configured that work group. Otherwise, see Configure a Work Group.

- You have already defined the TCP/IP Protocol that the host is to use. You can use the same protocol as defined when you configured the Component Broker cell and minimum work group. Otherwise, see “Define a TCP/IP or IPC Protocol” on page 170.
- You have already defined the Name Server that the host is to use. You can use the same name server as defined when you configured the Component Broker cell and minimum work group. Otherwise, see “Define a Name Server” on page 171.

**To configure a new host into your network, complete the following steps:**

1. Expand the Hosts Folder
2. On the pop-up menu of the Host, click **Drag**.
3. Expand the Configuration of the Management Zone that you are using to configure your host network.

   Configure the name server and TCP/IP protocol for the host:

   a. Expand the Name Servers folder

   b. On the pop-up menu of the Name Server, click **Configure Host**

   c. Expand the TCP/IP Protocols folder

   d. On the pop-up menu of the TCP/IP Protocol, click **Configure Host**

      **Note:** The **Configure Host** action was offered because the Host is still on the system management clipboard from the last **Drag** action.
4. Configure the host as preferring, and a member of, the Component Broker cell:

   a. Expand the Cells folder

   b. On the pop-up menu of the Cell, click **Configure Preferring Host**

   c. On the pop-up menu of the Cell, click **Configure Member Host**.
5. Configure the host as preferring, and a member of, one work group:

   a. Expand the Work Groups folder

   b. On the pop-up menu of one Work Group, click **Configure Preferring Host**

   c. On the pop-up menu of the same Work Group, click **Configure Member Host**.
6. If the host is to be a member of other work groups, configure the host as a member of those work groups.

   On the pop-up menu of each Work Group, click **Configure Member Host**.
7. Activate the Configuration, to check that the Configuration is valid and to add the host to the name tree.

### Related Concepts

“Hosts” on page 20
“The Model World” on page 39

### Related Tasks

“Configure the Component Broker Cell and Minimum Work Group” on page 111
“Configure a new Work Group” on page 115
“Configure a Host as a Member of Non-preferred Work Groups”

## Configure a Host as a Member of Non-preferred Work Groups

Use this procedure to add a host as a member of a work group other than the work group that it prefers.

You add a host to the work group that it prefers when you “Configure a new Host into your Network” on page 117. You cannot change the *Preferred Work Group* relationship of a host.

You can add a host as a member of one or more other work groups.

**Prerequisites:**

This task makes the following assumptions:

- You have already installed Component Broker onto the host. If not, see the *Planning, Performance, and Installation Guide* for information.
- You have already configured the host as a member of the Component Broker cell and the work group that it prefers. Otherwise, see “Configure a new Host into your Network” on page 117.

**To add a host to one or more work groups, complete the following steps:**

1. Expand the Hosts Folder
2. On the pop-up menu of the Host, click **Drag**.
3. Expand the Configuration of the Management Zone that you are using to configure your host network.
4. On the pop-up menu of each Work Group that the host is to be a member of, click **Configure Member Host**.
5. Activate the Configuration, to check that the Configuration is valid and to update the name tree.

**Related Concepts**

“Hosts” on page 20
“The Model World” on page 39
“Name Tree Configuration Management Rules” on page 168

**Related Tasks**

“Plan what to Configure for a Multi-Host Application Environment” on page 137
“Configure the Component Broker Cell and Minimum Work Group” on page 111
“Configure a new Host into your Network” on page 117
“Configure a new Work Group” on page 115

## Verify that a Name Server is in the DCE CDS

Use this task to verify that the name server on a host is properly stored in the DCE Director Cell Directory Service (CDS). *You should do this only after the name server has been configured and registered with DCE.*

To configure and register a host name server with DCE, you should have activated a Configuration of the Management Zone that defines your host network and from which you have configured a Name Server onto the Host. For more information, see the related tasks (page 120) at the end of this topic.

To verify that a host name server is properly stored in the DCE CDS domain, complete the following steps:

WIN

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. 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.name.com*-Name-Server) should be under servers in the tree view; where *myhost.name.com* is the long name for your host.

**AIX**

1. From a command prompt, enter:

   ```
   cdsbrowser
   ```
2. Double-click:

   ```
   /.:
   ```
3. 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&. long number) > host > resources > servers. The name server (*myhost.name.com*-Name-Server) should be under servers in the tree view; where *myhost.name.com* is the long name for your host.

**Note:** If you do not find the host name server listed within the DCE-CDS, check the following:

- The DCE application is correctly configured and is running
- You have configured a Name Server onto your Host in a Configuration of the Management Zone that you are using to define your Component Broker host network
- You have activated the Configuration of the Management Zone that your are using to define your Component Broker host network

Component Broker uses *myhost.name.com*-Name-Server as the default DCE principal for the host name server. If *myhost.name.com*-Name-Server already exists in the DCE Registry, the System Manager 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 host name server.

To find the value of the **security name** attribute, complete the following steps:

1. Start the System Manager user interface menu, and select (**View - User Level - Expert**)
2. Return to the Home view; for example by clicking the **Home** icon of the tool bar
3. Expand the Host Images folder
4. Expand the Host *myhost.name.com*
5. Expand the Server Images folder
6. On the pop-up menu of the Server Image *myhost.name.com*-Name-Server,click **Edit**. This opens the Object Editor.
7. Click the **Security Service** tab.

   The DCE principal of the name server is in the **security name** attribute.

**Related Concepts**

"Hosts" on page 20
"The Component Broker Name Tree and DCE" on page 23

**Related Tasks**

"Configure the Component Broker Cell and Minimum Work Group" on page 111
"Configure a new Work Group" on page 115
"Configure a new Host into your Network" on page 117

## Example: Configure a new Multi-Host Network

This topic describes how to configure a Component Broker multi-host network, based on the network shown in Figure 33 on page 122 so that the hosts are organized into the DCE cell used by Component Broker and into one or more work groups.

It provides an example of the general procedure described in "Configure a Multi-Host Environment" on page 109.

The aim is to guide you through organizing a new multi-host Component Broker network as easily as possible. To achieve this, the task steps are given with only the minimum background information to help you understand the reason for a step. If you need more information, see the related topics listed at the bottom of this page (page 126).

The task steps assume that you have not already activated a system management Configuration that defines the cell, work group, and serving hosts of your Component Broker host network. If you have done this, and want to reorganize your host network, see "Chapter 6. Administer your Host Environment" on page 167.

If you want to use the sample Management Zone provided with Component Broker, follow the steps given in the example, Example: Use the Sample Configuration for a Multi-Host Network instead of those listed below.

## An Overview of the Example Host Network

The example host network (shown in Figure 33 on page 122) comprises the following hosts, of which only the managed server hosts are organized into the Component Broker cell and work groups:

- System Manager host, host1.cbnet.com, does not run any application servers so is not part of the Component Broker cell or any work groups
- Managed server host, host2.cbnet.com, serves the Component Broker cell *cbcell* and the minimum work group *Work Group 1*
- Managed server host, host3.cbnet.com, a member of the Component Broker cell *cbcell* and work group *Work Group 1*
- Managed server host, host4.cbnet.com, a member of the Component Broker cell *cbcell* and *Work Group 2*, also serves work group *Work Group 2*
- Managed client-only host, host5.cbnet.com, does not run any application servers so is not part of the Component Broker cell or any work groups

**Note:** In this example host network, the System Manager user interface and the Java clients run on workstations that are not managed by Component Broker. However, those workstations are configured by Component Broker, to enable the System Manager user interface to communicate with the System Manager and to create appropriate client style properties files on the Java client workstations.

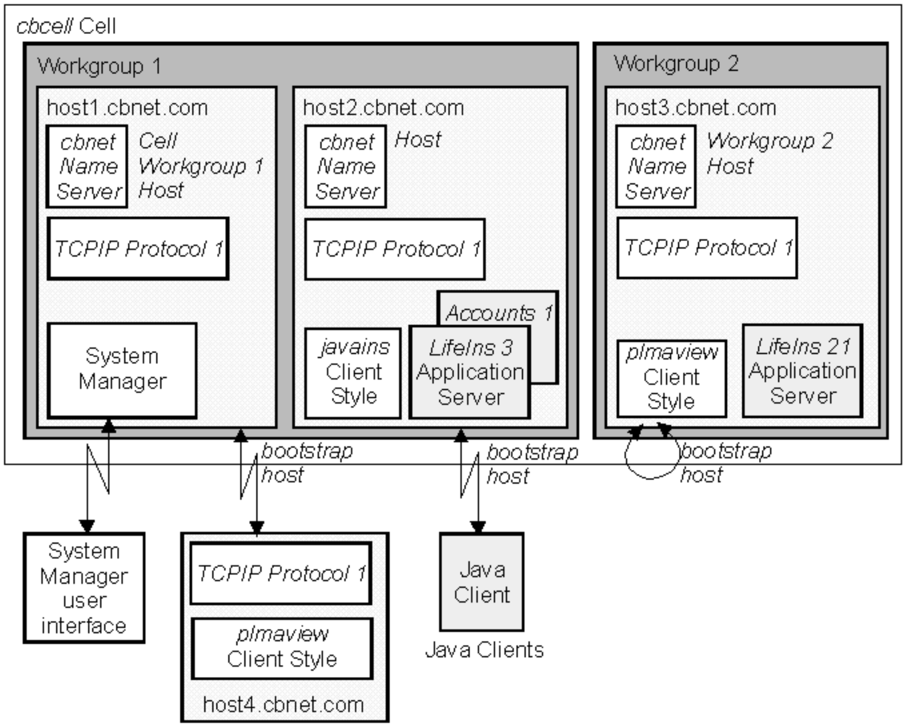


*Figure 33. The Example Host Network*

## The Task to Configure the Example Multi-Host Network

Use this task to configure the example multi-host network shown in Figure 33. The task creates and configures the system management objects listed in the following table. If you want to use the example to configure your own multi-host network, substitute the names listed with your own names.

*Table 6.* **System Management Objects Configured for the Example Multi-host Network**

| Object Type | Example Name | Your Name? |
|---|---|---|
| Management Zone | Cbroker Network Zone | |
| Configuration | Cbroker Config | |
| Cell | Cbcell | |
| Work Groups | Work Group 1<br>Work Group 2 | |
| Hosts | host2.cbnet.com<br>host3.cbnet.com<br>host4.cbnet.com | |
| Name Server | Cbnet | |
| TCP/IP Protocol | TCPIP Protocol 1 | |

**To configure the example multi-host network, complete the following steps:**

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Create the Management Zone for you to organize your Component Broker network.

   This Management Zone should be reserved only for organizing your Component Broker network into a cell and one or more work groups. You are recommended to use other Management Zones for your applications. You must activate the Configuration of the Management Zone created in this step before any of your application Management Zones.

   To create a new Management Zone, complete the following steps:

   a. On the pop-up menu of the Management Zones folder, click **Insert**.

      This opens a dialog box for you to specify a unique name for the new Management Zone.

   b. Type *CBroker Network Zone*

   c. To create the new Management Zone, click the **OK** button.

   d. To display the Management Zone, expand the Management Zones folder by clicking its + sign. You should see a Management Zone called *CBroker Network Zone*. If not, repeat the previous steps.
3. Create a Configuration within the Management Zone.

   Each Management Zone must contain at least one Configuration, which defines an implementation of the Management Zone.

   To create a new Configuration, complete the following steps:

   a. On the pop-up menu of the Management Zone *CBroker Network Zone*, click **New - Configuration**.

      This opens a dialog box for you to specify a unique name for the new Configuration.

   b. Type *CBroker Config*

   c. To create the new Configuration, click the **OK** button.

   d. To display the Configuration, expand the Configurations folder of the Management Zone *CBroker Network Zone*. You should see a Configuration called *CBroker Config*. If not, repeat the previous steps.
4. Expand the Configuration *CBroker Config*, by clicking its + sign.

   You create a Cell and one or more work groups within the Configuration to organize your Host network. Although you do not need to expand the Configuration at this point, it will help you to see the new folders and objects that you will create in the following steps. When you first create a Configuration, the object folders within it are empty and, by default, not visible.
5. Create the Cell. (The Component Broker network requires one, and can contain only one, cell.)

   a. On the pop-up menu of the Configuration *CBroker Config*, click **New - Cell**.

      This displays a dialog box for you to specify a unique name for the Cell.

   b. Type *Cbcell*

   c. To create the Cell, click the **OK** button.

   d. To display the Cell, expand the Cells folder. You should see a Cell called *Cbcell*. If not, repeat the preceding steps.
6. Create the minimum work group. (The Component Broker network requires at least one work group, which you use to group all hosts that you do not want in any other work group.)

   a. On the pop-up menu of the Configuration *CBroker Config*, click **New - Work Group**.

      This displays a dialog box for you to specify a unique name for the new Work Group.

   b. Type *Work Group 1*

   c. To create the Work Group, click the **OK** button.

   d. To display the Work Group, expand the Work Groups folder. You should see a Work Group called *Work Group 1*. If not, repeat the preceding steps.

7. Configure the host that serves the cell and minimum work group, host2.cbnet.com. Each host on which application servers are to run must have a TCP/IP Protocol and a Name Server configure on it.

   Define and configure the TCP/IP Protocol for the Host:

   a. On the pop-up menu of the Configuration *CBroker Config*, click **New - TCP/IP Protocol**.

      This displays a dialog box for you to specify a unique name for the new Protocol.

   b. Type *TCPIP Protocol*

      Note that the name cannot contain a forward slash character (as in *TCP/IP*).

   c. To create the TCP/IP Protocol, click the **OK** button.

   d. To display the Protocol, expand the TCP/IP Protocols folder. You should see an object called *TCPIP Protocol 1*. If not, repeat the preceding steps.

   e. Expand the Hosts folder

   f. On the pop-up menu of the Host *host2.cbnet.com*, click **Drag**.

   g. On the pop-up menu of the TCP/IP Protocol *TCPIP Protocol 1*, click **Configure Host**.

   You can configure the same TCP/IP Protocol onto other Hosts that need a protocol with the same characteristics, as described later.

   Define and configure the Name Server for the Host:

   a. On the pop-up menu of the Configuration *CBroker Config*, click **New - Name Server**.

      This displays a dialog box for you to specify a unique name for the new Name Server.

   b. Type *Cbnet*

   c. To create the Name Server, click the **OK** button.

   d. To display the Name Server, expand the Name Servers folder. You should see a Name Server called *Cbnet*. If not, repeat the preceding steps.

   e. On the pop-up menu of the Name Server, click **Configure Host**.

      **Note:** The **Configure Host** action was offered because the Host *host2.cbnet.com* is still on the system management clipboard from the last **Drag** action.

   You can configure the same Name Server onto other Hosts that need a name server with the same characteristics, as described later.

8. Define the host that serves the cell and minimum work group. (In a Component Broker network there is one host that serves both the cell and the minimum work group.)

The following steps assume that you have already expanded the Cells folder, and that last object dragged was the Host *host2.cbnet.com* that is to serve the cell. This is true if you followed the preceding steps.

a. On the pop-up menu of the Cell *Cbcell* click **Configure Serving Host**

b. On the pop-up menu of the Cell *Cbcell* click **Configure Preferring Host**

c. On the pop-up menu of the Cell *Cbcell* click **Configure Member Host**.

   The following steps assume that you have already expanded the Work Groups folder, and that last object dragged was the Host *host2.cbnet.com* that is to server the work group. This is true if you followed the preceding steps.

d. On the pop-up menu of the Work Group *Work Group 1* click **Configure Serving Host**

e. On the pop-up menu of the Work Group *Work Group 1* click **Configure Preferring Host**

f. On the pop-up menu of the Work Group *Work Group 1* click **Configure Member Host**.

   **Note:** If a host is to *ever* serve a work group, it must be defined by **Configure Serving Host** before *any* Configuration using the host is first activated.

9. Activate the Configuration. This creates the cell, minimum work group, and name server for the host. It also defines that the host name server is responsible for the runtime management of the cell and work group name tree.

   On the pop-up menu of the Configuration *CBroker Config*, click **Activate**.

   The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

10. To create the work group *Work Group 2*, and define that the host *host3.cbnet.com* serves that work group, complete the following steps:

    a. On the pop-up menu of the Configuration *CBroker Config*, click **New - Work Group**.

       This displays a dialog box for you to specify a unique name for the new Work Group.

    b. Type *Work Group 2*

    c. To create the Work Group, click the **OK** button. If the name is valid, this creates an object in the Work Groups folder. If the name is not valid, you are prompted to enter a new name.

    d. To display the Work Group, expand the Work Groups folder. You should see a Work Group called *Work Group 2*.

11. Configure the host that serves Work Group 2, *host3.cbnet.com*.

    (If host *host3.cbnet.com* is to ever serve a work group, it must be defined as serving a work group before any Configuration that uses that host is activated.)

    a. On the pop-up menu of the Host *host3.cbnet.com* click **Drag**

       Configure the name server and TCP/IP protocol for the host:

    b. On the pop-up menu of the Name Server *Cbnet*, click **Configure Host**

    c. On the pop-up menu of the TCP/IP Protocol *TCPIP Protocol 1*, click **Configure Host**

Configure the host as a member of, and preferring, the Component Broker cell:

d. On the pop-up menu of the Cell *Cbcell* click **Configure Preferring Host**

e. On the pop-up menu of the Cell *Cbcell* click **Configure Member Host**.

Configure the host as serving the new work group:

f. On the pop-up menu of the Work Group *Work Group 2* click **Configure Serving Host**

g. On the pop-up menu of the Work Group *Work Group 2* click **Configure Preferring Host**

h. On the pop-up menu of the Work Group *Work Group 2* click **Configure Member Host**.

12. Activate the Configuration again.

    On the pop-up menu of the Configuration *CBroker Config*, click **Activate**.

    Check the Action Console window for the *Activation successful* message.

13. Add the client-only host *host4.cbnet.com* to the host environment and configure the TCP/IP protocol for the host:

    a. On the pop-up menu of the Host click **Drag**.

    b. On the pop-up menu of the TCP/IP Protocol, click **Configure Host**

14. Activate the Configuration again.

    On the pop-up menu of the Configuration *CBroker Config*, click **Activate**.

    Check the Action Console window for the *Activation successful* message.

You have now defined the complete multi-host network shown in Figure 33 on page 122 so that the hosts are organized into the DCE cell used by Component Broker and into the two work groups.

The next stage is to create the application environment that is to use the host network, as described in “Example: Configure a new Multi-host Application Environment” on page 159. This adds the servers, client styles, and applications that are to run on the managed hosts.

Alternatively, you can change the organization of the host network; for example, by adding more hosts to work groups. For more information, see “Chapter 6. Administer your Host Environment” on page 167.

## Example: Use the Sample Configuration for a Multi-Host Environment

This topic describes how to use the samples provided with Component Broker to configure a multi-host environment, based on the host environment shown in the figure The Sample Host Environment. The tasks to be completed configure the hosts into the DCE cell and minimum work group required by Component Broker and into one or more other workgroups. They also configure the name server and TCP/IP protocol needed by the hosts.

It provides an example of the general procedure described in “Configure a Multi-Host Environment” on page 109.

Using the samples provided with Component Broker is an easy way to configure a host environment. However, for a full deployment multi-host environment you are likely to want to change the sample configuration to tailor it to the requirements of your own host environment For example, you are likelt to want to change the names of the sample system management objects to more meaningful names, change some of the objects’ attributes, and add to new objects required by your host environment.

The task steps assume that you have not already activated a system management Configuration that defines the cell, work group, and serving host of the Component Broker host environment. If you have done this, and want to reorganize your host environment, see

## An Overview of the Sample Host Environment

The sample host environment (shown in Figure 34) comprises the following hosts, of which only the managed server hosts are organized into the Component Broker cell and workgroups:

- System Manager host, host1.cbnet.com, serves the Component Broker cell *cbcell* and the minimum work group *Work Group 1*, but does not run any application servers
- Managed server host, host2.cbnet.com, is a member of the Component Broker cell *cbcell* and work group *Work Group 1*
- Managed server host, host3.cbnet.com, a member of the Component Broker cell *cbcell* and *Work Group 2*, also serves work group *Work Group 2*
- Managed client-only host, host4.cbnet.com, does not run any application servers so is not part of the Component Broker cell or any workgroups

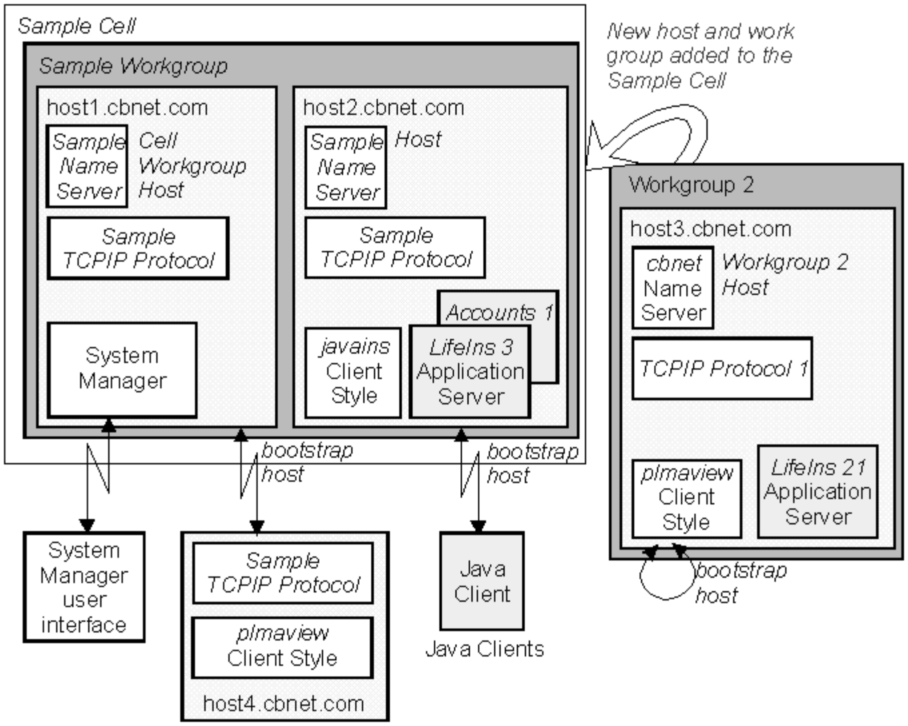


*Figure 34. The Sample Host Environment*

The following figure shows how the sample host environment is represented through the System Manager user interface.

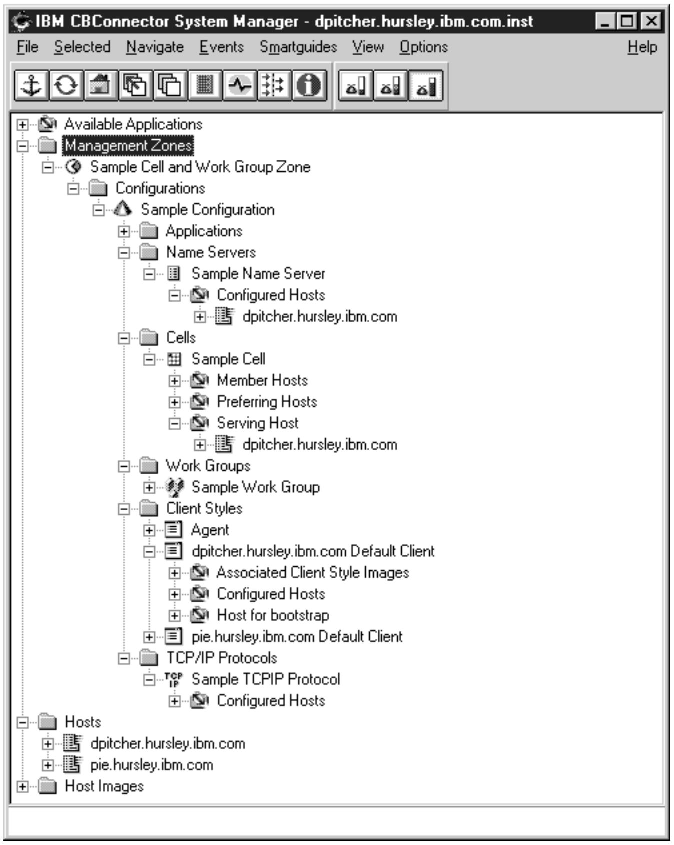


*Figure 35. The Sample Host Environment, as shown by the System Manager user interface*

## The Task to Configure the Sample Multi-Host Environment

Use this task to configure the sample multi-host environment shown in the figure Example: The Sample Host Environment. The task configures the system management objects listed in the following table. It also illustrates changing the names of the sample objects and adding new objects to the sample configuration. If you want to use the example to configure your own multi-host environment, substitute the names listed with your own names.

This example assumes that the host *host3.cbnet.com* has not been added to the host environment at first. It is added later in the example, to illustrate the addition and configuration of a new host and work group.

*Table 7.* **System Management Objects Configured for the Sample Multi-host Environment**

| Object Type | Sample Name | New Name | Your Name? |
|---|---|---|---|
| Management Zone | Sample Cell and Work Group Zone | Cbroker Network Zone | |
| Configuration | Sample Configuration | Cbroker Config | |
| Cell | Sample Cell | Cbcell | |
| Workgroups | Sample Work Group | Work Group 1<br>Work Group 2 | |

*Table 7.* **System Management Objects Configured for the Sample Multi-host Environment** *(continued)*

| Object Type | Sample Name | New Name | Your Name? |
|---|---|---|---|
| Hosts | host1.cbnet.com<br>host2.cbnet.com<br>host3.cbnet.com<br>host4.cbnet.com | n/a | |
| Name Server | Sample Name Server | Cbnet | |
| TCP/IP Protocol | Sample TCPIP Protocol | TCPIP Protocol 1 | |

**Note:** Besides the system management obects listed above, the Sample Configuration contains a default client style (called *hostname* Default Client Style) for the host and some default applications for Component Broker services. You should not rename those system management objects.

**To configure the example multi-host network, complete the following steps:**

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Expand the Management Zones folder
3. Expand *Sample Cell and Work Group Zone*. To rename the sample Management Zone, complete the following steps:
   a. On the pop-up menu of *Sample Cell and Work Group Zone*, click **Rename**. This displays a dialog box for you to specify a unique name for the Management Zone.
   b. Type *CBroker Network Zone*
   c. To apply the change, click the **OK** button.
4. Expand the Configurations folder
5. Expand *Sample Configuration*. To rename the Sample Configuration, complete the following steps:
   a. On the pop-up menu of *Sample Configuration*, click **Rename**. This displays a dialog box for you to specify a unique name for the Configuration.
   b. Type *CBroker Config*
   c. To apply the change, click the **OK** button.
6. To display the sample Name Server, expand the Name Servers folder. To rename the Name Server, complete the following steps:
   a. On the pop-up menu of *Sample Name Server*, click **Rename**. This displays a dialog box for you to specify a unique name for the Name Server.
   b. Type *Cbnet*
   c. To apply the change, click the **OK** button.
7. To display the sample Cell, expand the Cells folder. To rename the sample Cell, complete the following steps:
   a. On the pop-up menu of *Sample Cell*, click **Rename**. This displays a dialog box for you to specify a unique name for the Cell.
   b. Type *Cbcell*
   c. To apply the change, click the **OK** button.
8. To display the sample Work Group, expand the Work Groups folder. To rename the sample Work Group, complete the following steps:
   a. On the pop-up menu of *Sample Work Group*, click **Rename**. This displays a dialog box for you to specify a unique name for the Work Group.
   b. Type *Work Group 1*

   c. To apply the change, click the **OK** button.

9. To display the sample TCP/IP Protocol, expand the TCP/IP Protocols folder. To rename the sample TCP/IP Protocol, complete the following steps:
   a. On the pop-up menu of *Sample TCPIP Protocol*, click **Rename**. This displays a dialog box for you to specify a unique name for the Protocol.
   b. Type *TCPIP Protocol*. Note that the name cannot contain a forward slash character (as in *TCP/IP*).
   c. To apply the change, click the **OK** button.
10. Activate the Configuration. This creates the cell, minimum work group, and name servers. It also defines that the name server on *host1.cbnet.com* is responsible for the runtime management of the cell and work group name tree.

    On the pop-up menu of the Configuration *CBroker Config*, click **Activate**.

    The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have installed Component Broker properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, you should see a *Activation successful* message.
11. To add the work group *Work Group 2*, and define that the host *host3.cbnet.com* serves that work group, complete the following steps:
    a. Install Component Broker onto host3.cbnet.com, using the **Application Server** installation option. This creates a Host object called host3.cbnet.com. (This step is beyond the scope of this example. For instructions and examples about installing Component Broker, see the *Planning, Performance, and Installation Guide.)*
    b. On the pop-up menu of the Configuration *CBroker Config*, click **New - Work Group**. This displays a dialog box for you to specify a unique name for the new Work Group.
    c. Type *Work Group 2*
    d. To create the Work Group, click the **OK** button. If the name is valid, this creates an object in the Workgroups folder. If the name is not valid, you are prompted to enter a new name.
    e. To display the Work Group, expand the Workgroups folder. You should see a Work Group called *Work Group 2*.
    f. Add the *host3.cbnet.com* to the host environment and configure it to serve Work Group 2.

       (If host *host3.cbnet.com* is to ever serve a work group, it must be defined as serving a work group before any Configuration that uses that host is activated.)

       1) On the pop-up menu of the Host *host3.cbnet.com* click **Drag**

          Configure the name server and TCP/IP protocol for the host:

       2) On the pop-up menu of the Name Server *Cbnet*, click **Configure Host**
       3) On the pop-up menu of the TCP/IP Protocol *TCPIP Protocol 1*, click **Configure Host**

          Configure the host as a member of, and preferring, the Component Broker cell:

       4) On the pop-up menu of the Cell *Cbcell* click **Configure Preferring Host**
       5) On the pop-up menu of the Cell *Cbcell* click **Configure Member Host**.

          Configure the host as serving the new work group:

6) On the pop-up menu of the Work Group *Work Group 2* click **Configure Serving Host**

7) On the pop-up menu of the Work Group *Work Group 2* click **Configure Preferring Host**

8) On the pop-up menu of the Work Group *Work Group 2* click **Configure Member Host**.

g. Activate the Configuration again.

On the pop-up menu of the Configuration *CBroker Config*, click **Activate**.

Check the Action Console window for the *Activation successful* message.

You have now defined the complete multi-host environment shown in Figure 33 on page 122 so that the hosts are organized into the DCE cell used by Component Broker and into the two workgroups.

The next stage is to create the application environment that is to use the host environment, as described in Example: Configure the Sample Application Environment. This adds the servers, client styles, and applications that are to run on the managed hosts.

Alternatively, you can change the organization of the host environment; for example, by adding more hosts to workgroups. For more information, see "Chapter 6. Administer your Host Environment" on page 167.

**To rename a system management object:** To change the name of any system management object, you use the following procedure:

1) On the pop-up menu of the object, click **Rename**

2) In the Object Rename dialog window, type the new name for the object

3) To apply the change, click the **OK** button.

# Chapter 5. Configure a new Application Environment

This topic describes how to configure a new application environment to be managed by Component Broker.

The tasks to be completed, and application environment that is created, apply equally to both single- and multi-host Component Broker enterprises. The term *System Manager host* refers to the host on which the System Manager runs; for a single-host environment, that is the one host in that environment.

The complete task of configuring an application environment is split into several smaller tasks, to help you develop your application environment more easily in several stages. Some of these tasks can also be used at a later time to add servers, client styles, and applications to your application environment.

An example is provided at the end of this topic.

The task steps assume that you have already activated a system management Configuration that defines the cell, work group, and serving hosts of your Component Broker host environment.

If you need more information about administering your application environment, see the related topics listed at the bottom of this page (page 134).

## An Overview of Configuring an Application Environment

The general procedure to configure a new application environment involves the following tasks:

1. Create a new Management Zone and, within that, a Configuration to contain your application environment.

   You should use a different Management Zone to the one that you used to configure your host environment into the Component Broker cell and one or more workgroups. You must activate the Configuration of the application Management Zone created in this task *after* you have activated your host environment Management Zone.
2. To enable you to configure an application to be managed by Component Broker, you should install the application family package onto the System Manager host. This installs the application software onto that host and completes the initial configuration of the application and its resources. Normally you install an application using the installation tool provided with the application family package.

   This task also involves loading the application into a Configuration, to enable the application to be configured onto server groups, free standing servers, or client styles as appropriate. If required, change the default properties of the application by performing additional configuration of the application and its resources *as instructed by information provided with the application family package*. The task ends with the application loaded into a Configuration and its properties and resources configured as required.
3. If you want to run an application on several servers, it is usually simplest to configure the servers as members of a server group. You define a *Server*

*Group*, configure onto it the applications to run on servers in the group, and if needed edit it to change the default attributes of all the *Servers (member of group)*.

This task also involves configuring each server in the group onto the host that it is to run on. The task ends with activating the Configuration, to check that the Configuration is valid, start the servers on their hosts, and start the applications on the servers.

4. If you want to run an application on a server with unique characteristics, or only want one application server in your application environment, you can define the server as a *Server (freestanding)*. You use the Server (freestanding) to configure the applications to run on the server and if needed to change the default attributes of the server.

   This task also involves configuring the server onto the host that it is to run on. The task ends with activating the Configuration, to check that the Configuration is valid, start the server on its host, and start the applications on the server.

5. Component Broker automatically creates a default client style. You can also define you own client styles to cater for clients with special properties. You should define a client style for each type of client managed by Component Broker and for each type of Java client whose properties file is to be configured by Component Broker.

   This task also involves configuring the client style onto the hosts that it is to be available on and, if needed, configuring client applications to run on the client style. The task ends with activating the Configuration, to check that the Configuration is valid, make the client style available on appropriate hosts, and create properties files needed by Java clients. An additional closing step is to copy the Java properties files to the workstations and web servers that need them.

To configure a new application environment, complete the tasks described in “Configure a new Application Environment” on page 139.

**Related Tasks**

“Chapter 7. Administer Application Servers” on page 181
“Chapter 8. Administer Clients” on page 205
“Chapter 9. Administer Applications” on page 221
“Chapter 10. Administer Management Zones and Configurations” on page 253
“Chapter 11. Administer Security in your Enterprise” on page 261
“Chapter 12. Administer Workload Management” on page 333
“Chapter 13. Administer Connections to Tier-3 Systems” on page 343

## Plan what to Configure for a Single-Host Application Environment

Before configuring your application environment, you should create a plan of the environment and identify the components that you need to configure. The decisions that you make when planning your application environment are generally dictated by the applications that you want in your enterprise and the information provided with those applications.

The points to consider, as described in this topic, have been tailored to configuring an application environment to run on a standalone host. The same points apply in a multi-host application environment, but are tailored to such an environment in “Plan what to Configure for a Multi-Host Application Environment” on page 137.

Consider the following points, based on the general single-host application environment shown in Figure 36 on page 136. From these points, you should be able to create a plan of the following components to configure, like that given in the table Components to Configure for a Single-host Application Environment (page Figure 36 on page 136) after the figure.

**Server Groups and Servers (member of group)**

You can configure an application onto one or more *server groups*. Each server group contains one or more member application servers with the same characteristics. You define a *Server Group* to set the characteristics of all the servers in the group and to configure applications for the server group. You identify the servers in the group by *Servers (member of group)*.

**Freestanding Servers**

If you only need one application server, or need an application server with unique characteristics, you can define one as a *Server (freestanding)*. You use the Server (freestanding) to set the characteristics of the server, configure the applications it is to run, and configure the server onto the host that it is to run on.

At any time, you can convert a freestanding server into a server group or convert a server (member of group) into a freestanding server.

**Client Styles**

You can manage the properties of clients through one or more *Client Styles*, which you configure onto the host. For managed clients, the SM agent on a host uses the client style to dynamically change the properties of the clients. For Java clients, a client style is used to configure a Java properties file that you copy either onto the client workstation or, more often, onto a Web Server accessible from the client.

Component Broker provides a default client style for managed clients on the host.

**Applications**

Applications are normally provided as part of an *application family* package and installed onto the System Manager host by the application installation tool provided with the package. An *application family* contains one or more DDL files that automatically define the applications and their resources for system management when you install the applications. The application family package should provide information about any changes you need to make to the properties of applications or the resources they need after installation. Otherwise, you need only configure an application onto the server group, freestanding server, or client style on which it is to run.

**Management Zone and Configuration**

You configure your application environment within Configurations of one or more application Management Zones. You are recommended to use application Management Zones only for configuring your application environment, and use a separate Management Zone for configuring the host environment. You must activate the Configuration of the Management Zone used for your network before any of your application Management Zones.

You can configure all your application environment in one Management Zone. However, you can separate the administration of parts of your application environment into several Management Zones. Each application Management Zone can use unique hosts within your host network or can use the same hosts as other application Management Zones.

You can define alternative Configurations of each application Management Zone to cater for repeated changes in your application environment.

The above points are only for the basic functionality of an application environment. You can extend that functionality as described in the related topics; for example, to enable security, workload management, or communication with tier-3 systems.

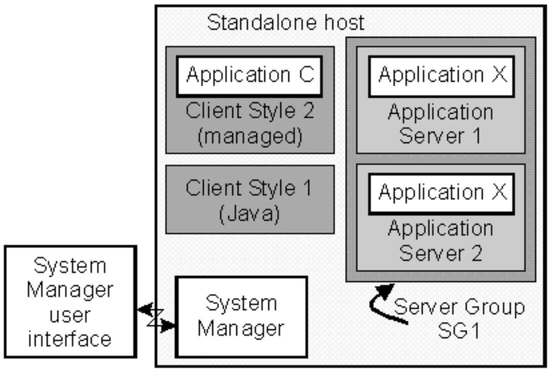


*Figure 36. A General Single-host Application Environment*

*Table 8.* **Components to Configure for a Basic Single-host Application Environment**

| Component | Name | Your value? | Applications Run |
|---|---|---|---|
| **Server Group** | SG1 | | X ... |
| **Servers (member of group)** | AS 1<br>AS 2 | | n/a (defined on Server Group) |
| **Client Style (Java)** | CS 1 | | |
| **Client Style (managed)** | CS 2 | | C ... |
| **Management Zone** | zone_name | | n/a |
| **Configuration** | config_name | | n/a |

*Table 9.* **Java Clients: Target Hosts/Workstations for Properties File**

| | Name | Your value? |
|---|---|---|
| **Client Style (Java)** | CS 1 | |
| **On Host** | host1.cbnet.com | |
| **Targets for Properties File** | Workstations j1 through j20, Web server www.s1.com | |

If you want to proceed with configuring a new single-host application environment, see Configure a Single-host Application Environment.

For an example of configuring a new single-host application environment, see “Example: Configure a new Single-host Application Environment” on page 154.

## Plan what to Configure for a Multi-Host Application Environment

Before configuring your application environment, you should create a plan of the environment and identify the components that you need to configure. The decisions that you make when planning your application environment are generally dictated by the applications that you want in your enterprise and the information provided with those applications. The decisions also influence, and are influenced by, the structure of your Component Broker host network.

Consider the following points, based on the general multi-host application environment shown in Figure 37 on page 138. From these points, you should be able to create a plan of the following components to configure, like that given in the table Components to Configure for a Basic Application Environment (page 138) after the figure.

**Server Groups and Servers (member of group)**

You can configure an application onto one or more *server groups*. Each server group contains one or more member application servers with the same characteristics. You define a *Server Group* to set the characteristics of all the servers in the group and to configure applications for the server group. The servers in the group, which you define by *Servers (member of group)* are typically distributed across several hosts in your host network.

**Freestanding Servers**

If you only need one application server, or need an application server with unique characteristics, you can define one as a *Server (freestanding)*. You use the Server (freestanding) to set the characteristics of the server, configure the applications it is to run, and configure the server onto one host in your host network.

At any time, you can convert a freestanding server into a server group or convert a server (member of group) into a freestanding server.

**Client Styles**

You can manage the properties of clients through one or more *Client Styles*, which you configure onto one or more hosts in your host network. For managed clients, the SM agent on a host uses the client style to dynamically change the properties of the clients. For Java clients, a client style is used to configure a Java properties file that you copy either onto the client workstation or, more often, onto a Web Server accessible from the client.

**Applications**

Applications are normally provided as part of an *application family* package and installed onto the System Manager host by the application installation tool provided with the package. An *application family* contains one or more DDL files that automatically define the applications and their resources for system management when you install the applications. The application family package should provide information about any changes you need to make to the properties of applications or the resources they need after installation. Otherwise, you need only configure an application onto the server group, freestanding server, or client style on which it is to run.

**Management Zone and Configuration**

You configure your application environment within Configurations of one or more application Management Zones. You are recommended to use application Management Zones only for configuring your application environment, and use a separate Management Zone for configuring your host network. You must activate the Configuration of the Management Zone used for your network before any of your application Management Zones.

You can configure all your application environment in one Management Zone. However, you can separate the administration of parts of your application environment into several Management Zones. Each application Management Zone can use unique hosts within your host network or can use the same hosts as other application Management Zones.

You can define alternative Configurations of each application Management Zone to cater for repeated changes in your application environment.

The above points are only for the basic functionality of an application environment. You can extend that functionality as described in the related topics; for example, to enable security, workload management, or communication with tier-3 systems.

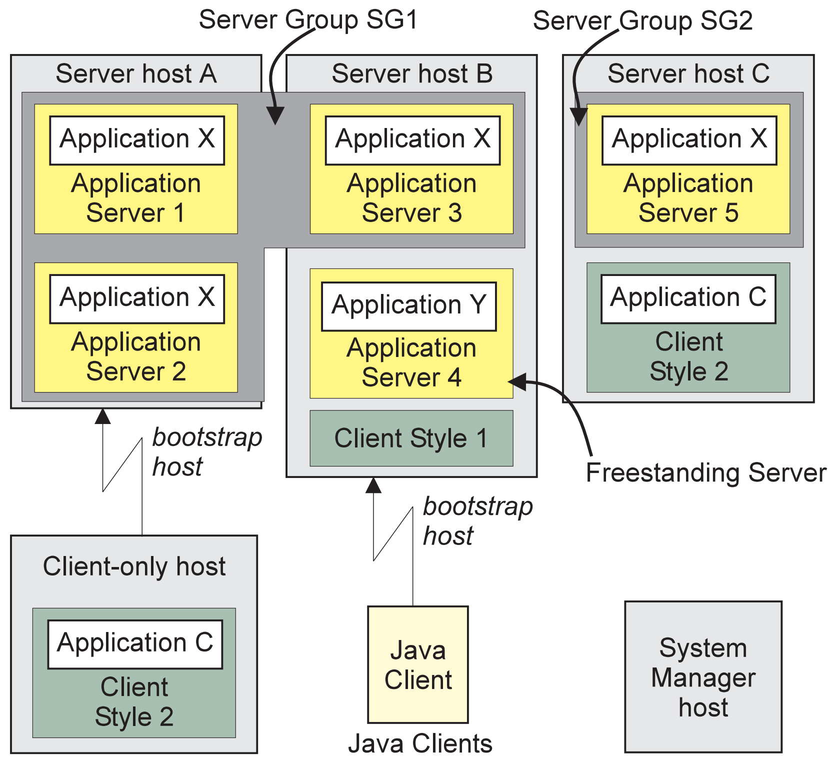


*Figure 37. A General Multi-Host Application Environment*

*Table 10.* **Components to Configure for a Basic Application Environment**

| Component | Name | On Host | Applications Run |
|---|---|---|---|
| **Server Group** | SG1 | | X ... |
| **Servers (member of group)** | AS 1<br>AS 2<br>AS 3 | Host A<br>Host A<br>Host B | |
| **Server Group** | SG2 | | X ... |
| **Servers (member of group)** | AS 5 | Host C | |
| **Freestanding Server** | AS 4 | Host B | Y ... |

*Table 10.* **Components to Configure for a Basic Application Environment** *(continued)*

| Component | Name | On Host | Applications Run |
|---|---|---|---|
| **Client Style (Java)** | CS 1 | Host B | |
| **Client Style (managed)** | CS 2 | Host C<br>Client Host | C ... |
| **Management Zone** | zone_name | | |
| **Configuration** | config_name | | |

*Table 11.* **Java Clients**

| Client Style (Java) | On Host | Targets for Properties File |
|---|---|---|
| CS 1 | Host B | Workstations j1 through j20, Web server www.s1.com |

If you want to proceed with configuring a new multi-host application environment, see “Chapter 5. Configure a new Application Environment” on page 133.

For an example of configuring a new multi-host application environment, see “Example: Configure a new Multi-host Application Environment” on page 159.

# Configure a new Application Environment

To configure a new application environment, complete the steps given in the following topics. Each topic guides you through to the next topic.

1. “Create a new Management Zone and Configuration for your Application Environment”
2. “Load a new Application” on page 141
3. If required, “Configure a new Server Group” on page 145
4. If required, “Configure a new Freestanding Server” on page 149
5. If required, “Configure a new Client Style” on page 151

For an example of the complete steps to configure a new application environment, see one of the following topics:

- “Example: Configure a new Single-host Application Environment” on page 154.
- “Example: Configure a new Multi-host Application Environment” on page 159.

## Create a new Management Zone and Configuration for your Application Environment

Complete the steps in this topic to create a new application Management Zone and, within that, a Configuration to be used to configure your application environment.

You are recommended to use application Management Zones only for configuring your application environment, and use a separate Management Zone for configuring your host network. You must activate the Configuration of the Management Zone used for your network before any of your application Management Zones.

You can configure all your application environment in one Management Zone. However, you can separate the administration of parts of your application environment into several Management Zones. Each application Management Zone can use unique hosts within your host network or can use the same hosts as other application Management Zones.

You can define alternative Configurations of each application Management Zone to cater for repeated changes in your application environment.

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. To create a new Management Zone, complete the following steps:
   a. On the pop-up menu of the Management Zones folder, click **Insert**

      This opens a dialog box for you to specify a unique name for the new Management Zone. Type the name that you want the new object to be known by. The name can contain:
      - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
      - Embedded blanks

   b. To create the new Management Zone, click the **OK** button. If you specified a valid name, this creates a new Management Zone. If the name specified is not valid, a dialog box is displayed for you to enter a new name.
3. Expand the Management Zones folder, by clicking its + sign. You should see an object for the Management Zone that you created. If not, repeat the previous step.
4. Create a Configuration within your Management Zone.

   Each Management Zone must contain at least one Configuration, which defines an implementation of the Management Zone.

   a. On the pop-up menu of your Management Zone, click **New - Configuration**.

      This opens a dialog box for you to specify a unique name for the new Configuration. Type the name that you want the new object to be known by. The name can contain:
      - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
      - Embedded blanks

   b. To create the new Configuration, click the **OK** button. If you specified a valid name, this creates a new Configuration. If the name specified is not valid, you are prompted to enter a new name.

   c. Expand the Configurations folder of your Management Zone. You should see an object for the Configuration that you created. If not, repeat the previous steps.

Next “Load a new Application” on page 141.

### Related Concepts

“Management Zones” on page 17
“Configurations” on page 18
“Application Servers and Server Groups” on page 25
“Clients” on page 207
“Applications” on page 27

### Related Tasks

“Create a new Management Zone and Configuration for your Host Network” on page 110

page 110
“Add a new Configuration to a Management Zone for your Application Environment” on page 254

## Load a new Application

Use this procedure to install an application into Component Broker System Management.

This procedure copies the application files to a unique directory under Component Broker and creates the system management objects used to represent and manage the application. It also adds the application to a Configuration of a Management Zone that you have created for your application environment. This enables you to later configure the application onto servers or client styles and enables the System Manager to automatically copy application files to where they are needed.

Normally, this procedure is completed automatically by the application installation tool. However, the procedure can be completed using the **Load Application** action through the System Manager user interface. **The Load Application action is normally used only in an application development environment before a proper tool is available to install an application.**

**Before starting this procedure, consider the following points:**

- You must load an application onto the System Manager’s host, so that it is aware of the available application. The System Manager can later distribute automatically the application to other hosts according to the Configurations that you create. This makes it easier for you to manage the installation, distribution, and uninstallation (if needed) of applications in a multi-host enterprise.
- If you use the **Load Application** action, you will need to specify the full path name of the application’s DDL file (in the /bin subdirectory of the application family). The System Manager uses the application DDL file to identify the system management objects to be created for the application and the location of the application’s files to be loaded. The name of the DDL file is also used to name the directory, *cbinstall/apps/familyname*, into which the application files are to be copied, where:
  - *cbinstall* is the name of the directory into which Component broker was installed
  - *apps* is the name of the application installation directory for the platform on which the application is to run. Applications to run on Windows NT are installed into *cbinstall/NTApps/familyname*. Applications to run on AIX are installed into *cbinstall/AIXApps/familyname*.
  - *familyname* is the name of the application DDL file

  If your application was created to use queriable objects, it will have a second *specific*family.ddl* file, which has the information needed to do queries. You should use the **Load Application** action, with the *specific*family.ddl* file, to add the extra information to the system management objects created for the application.
- By default, application servers do not have the functions (for example, homes) to support the creation of lifeCycle, event, or notification objects. If your application wants to use any of these services, you must add the service support needed for those objects onto one of the servers in your system management Configuration, as described in “Configure a Server to Provide Extended Component Broker

Services” on page 406. Note that you can configure the service onto any server that you want to provide these services, and an application in any other server can make use of those services.

- You should already have created the Configuration of the Management Zone for your application environment, as described in “Create a new Management Zone and Configuration for your Application Environment” on page 139

**To install an application using its own installation tool, complete the following steps:**

1. Insert the application compact disk *into the System Manager’s host computer*
2. Follow the instructions displayed by the application installation tool to specify parameters required by the application.
3. The application installation tool copies the application files to the application family directory and configures the application for system management.
4. You should then add the application into a Configuration of your application Management Zone.

**To install an application using the Load Application action, complete the following steps:**

1. Start the System Manager User Interface and set the User Level to Expert. (Click **View - User Level - Expert** .)
2. Expand the Host Images folder, to display the icons for managed hosts.
3. From the pop-up menu of the System Manager’s host, click **Load Application**.

   (If you are unsure which icon is for the System Manager’s host, the name of the System Manager is displayed on the title bar of the Information Controller window.)
4. In the entry field of the *Load Application* dialog window, type the full path name of the application’s DDL file.

   You can use the **browse** button to search for and select the application’s DDL files.

   **Notes:**

   a. The browse dialog box lists directories visible on the workstation ( WIN ) or host ( AIX ) on which the System Manager user interface is running, regardless of which Host Image you select the action on.
   b. ( WIN ) To load a DDL file on a remote host (for example, if the System Manager is running on a remote host), you can map the network drive into which the application was installed on that remote host. Make sure that the account you use to map the network drive has the same permissions as the account under which you are running the CBConnector service.
5. The Load Application action copies the application files to the application family directory and configures the application for CBConnector System Management.

   **Notes:**

   - You do not need to add the dlls and jar files to the PATH / CLASSPATH environment variables respectively:
     - DLLs are loaded automatically by Component Broker, using the fully qualified name; for example, c:\cbroker\ntApps\myapp\myappinit.dll
     - The classpath for the application server is also set automatically to contain the .jar file for the JAVABO’s/PAOs that the application uses; for example, c:\cbroker\ntApps\myapp\myappjar.jar

- When an application server first starts a Java virtual machine, the effective classpath is printed in the activity.log. You can view the information by formatting the log.
- To change application environment variables used by the CbConnector service, you can use the following actions:
  - Update the user environment variables for the userid that the CBConnector service is running under, then stop and restart the service.
  - Some Environment variables (such as SYSTEM CLASSPATH) are not refreshed for services until you reboot the host computer.
  - After rebooting a host, application servers that need to see the new classpath update must be reactivated. That is, do not simply start the server. You must activate the system management Configuration that contains the server.

6. You can monitor the progress of the **Load Application** action in its Action Console window. If the action completes successfully, the Action Console window displays a success statement. If the action fails, the Action Console window displays messages to indicate the cause of the problem. For example, if you see a message like ″... Could not open file x:\path\AppFam.ddl″, the account that you use to map the network drive may not have the same permissions as the account under which you are running the CBConnector service.

   When the **Load Application** action has completed successfully, reset the User Level to your normal level.
7. You should then add the application into a Configuration of your application Management Zone.

**To add the new application into a Configuration of your application environment, complete the following steps:**

1. Scroll to the top of the View panel
2. Expand the Available Applications folder You should see an Application with the name of the application that you just loaded.
3. On the pop-up menu of the Application, click **Drag**
4. Display the Configuration in which you want the Application to be created
5. On the pop-up menu of the Configuration, click **Add Application**

   This automatically creates the Application with the attributes and relationships needed by the application, and creates other objects needed by the application (in particular; Database Aliases, but also Containers, C++ Classes, and so on).
6. Optionally, to verify that the Application has been created, complete the following steps:
   a. Expand the Configuration on which you selected the **Add Configuration** action.
   b. Expand the Applications folder. You should see an object with the same name as the Application Install object selected in 2. If not, repeat steps 2 to 4.
   c. Optionally, to display any system management objects created for the application resources, expand the Application and its *provides ...* relationship folders.

After you have defined an Application, you probably next want to do one or more of the following:

- Edit the Application or its resource objects as needed by the application. For information about editing model objects for an application, see the information provided by the application. (See also “Edit Objects” on page 72.)
- Configure a new server group on which the application is to run, as described in “Configure a new Server Group” on page 145.
- Configure a new freestanding server on which the application is to run, as described in “Configure a new Freestanding Server” on page 149.
- Configure a new client style on which the application is to run, as described in “Configure a new Client Style” on page 151.
- Configure the Application onto one or more existing Server Groups, Servers (free standing), or Client Styles as required for your application environment.

**Note:** Several applications can be processed at the same time, by the following steps:

1. Anchor the Information Controller window
2. On the pop-up menu of the Available Applications folder, click **open**. This opens the folder in a new Information Controller window.
3. On the menu bar of the new window, click **View - View Type - List View**
4. Select several applications.
5. Click **Selected - Drag**
6. In your original window, on the pop-up menu of the Configuration of your application Management Zone, click **Add Application**

   This displays a message window for you to confirm that you want to add all the applications to your Configuration.
7. To continue, click the **Yes** button in the message window. Otherwise, click the **No** button.
8. Optionally, to verify that the Applications have been created, expand the Applications folder. You should see objects with the same names as the Application Installs selected in step 4. If not, repeat steps 4 to 6.
9. Optionally, close the new window and unanchor your original window.

### Related Concepts

“DDL Files for Applications” on page 13
“Installing Applications” on page 222
“The Install World” on page 41
“Applications” on page 27

### Related Tasks

“Add an Application into a Configuration of your Application Environment” on page 228
“Configure an Application onto a Freestanding Server” on page 186
“Add a Client Application into a Configuration of your Application Environment” on page 230
“Configure Applications onto a Client Style” on page 213
“Uninstall an Application” on page 224
“Refresh an Application” on page 225
“Configure a Server to Provide Extended Component Broker Services” on page 406

## Configure a new Server Group

Use this task to configure a new server group in a Configuration of one of your application Management Zones.

**Considerations:**

- Before you first activate a Configuration, you should decide whether each Server Group that it contains is to be always a controlled server group (for workload management) or basic server group, and configure the Server Group accordingly. (You should not normally later switch any server group from its original controlled or basic configuration.)
- The applications that are to run on members of the server group are configured onto the Server Group. You can do this when configuring a new server group as described in this topic, or later as described in “Configure Applications onto a Server Group” on page 183.
- AIX If an application server needs to use shared libraries that are not in one of the directories specified in the LIBPATH, you must make those libraries available before the application server is started. For information about how to do this, see “Make AIX Shared Libraries Available to Application Servers” on page 191.
- If required, after completing the procedure in this topic, you should change the default characteristics of the server group. You can do this by editing the attributes of the Server Group and perhaps configuring other system management objects onto it. This is normally dictated by the special requirements of an application, and by detailed information provided with the application family package. Some typical tasks are also provided in related task topics.

  The attribute values for the Server Group are used for all servers that are defined as members of the server group. Those servers are defined by the Servers (member of group) within the Server Group.

**To configure a new server group in your application environment**, you can use the **Create Servers** and **Configure Servers** wizards, to complete the following main stages:

1. 145
2. Configure one or more applications onto the server group for those applications to run on members of the server group (page 147)
3. Activate the Configuration (page 148), to check that Configuration is valid and to create the servers in the server group on their hosts. This also automatically copies the application files needed to the server hosts.

You can also use this task for an existing server group, to 145 or configure new applications onto the server group (page 147).

**To define the server group and the servers that are members of the server group, complete the following steps:**

1. On the menu-bar of the System Manager user interface, click **Wizards - Create Servers**
2. On the Management Zone panel, select the application Management Zone within which you want to create the server group. If this panel does not list an

appropriate Management Zone, you can type the name of a new Management Zone, which will be created automatically by the wizard.

3. Click the **Next** button
4. On the Configuration panel, select the Configuration (of the Management Zone that you selected previously) within which you want to create the server group. If this panel does not list an appropriate Configuration, you can type the name of a new Configuration, which will be created automatically by the wizard.
5. Click the **Next** button
6. On the Server Group panel, select the name of the server group. You can either select the name of an existing server group, displayed in the list box, or type the name for a new server group. (See Names of Server Groups and Servers.)

   If you want to configure a controlled server group for workload management, complete the following steps:

   a. Select the tick box for a **Configure Managing Host?**
   b. Select the host on which the server group's control point (SGCP) and gateway (SGGW) servers are to run.

   If you want the server group to ever be a controlled server group, you should select this option when first creating the server group. However, if you choose not to do this now, it is possible to reconfigure the server group as a controlled server group later.

   If the server group was previously configured as a controlled server group, you cannot change the host on which the SGCP server and SGGW server are to run.
7. Click the **Next** button
8. On the Servers panel, select the name of the application servers that are to be members of the server group.

   To add a server to the group, displayed in the list box, complete the following steps:

   a. Type a unique name for the new server. (See Names of Server Groups and Servers.)
   b. Select the host on which the server is to run.
   c. Click the **add** button

   The server, and its host, are added to the server group's list box.

   To remove a server from the group, displayed in the list box, complete the following steps:

   a. Select the server's entry in the server group's list box.
   b. Click the **remove** button

   The server, and its host, are removed from the server group's list box.
9. To finish and have the Create Server wizard create the server group, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

   If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

**Note:** The wizard does not create or configure what you have selected until you click the **finish** button on any panel and have specified the details that it needs. This defines the server group within the Configuration specified, but does not add the servers to your enterprise.

The server group is added to your application environment the next time that you activate the Configuration.

**To configure one or more applications onto the server group, complete the following steps:**

1. On the menu-bar of the System Manager user interface, click **Wizards - Configure Servers**.
2. On the Select Applications to Configure panel, select the applications that you want to configure onto the server group.

   To add an application to the list to be configured, displayed under *Applications to Configure*, complete the following steps:

   a. In the *Available Applications* list box, click on one or more applications. To select several applications, press and hold the Ctrl key while clicking on the applications then release the Ctrl key.

   b. Click the **Add** button.

   The applications are added to the list of applications to be configured on the server group, listed in the *Applications to Configure* list box.

   To remove an application from the *Applications to Configure* list box, complete the following steps:

   a. Select the application’s entry in the *Applications to Configure* list box.

   b. Click the **Remove** button.

   The application is removed from the list of applications to be configured on the server group.
3. Click the **Next** button
4. On the Management Zone panel, select the application Management Zone within which the server group is defined.
5. On the Configuration panel, select the Configuration (of the Management Zone that you selected previously) within which the server group is defined.
6. Click the **Next** button
7. On the Server Group panel, select the name of one or more server groups onto which the applications are to be configured.

   To add a server group to the list under *Servers To Configure Applications On*, complete the following steps:

   a. In the *Available Server Groups* list box, click on one or more server groups. To select several server groups, press and hold the Ctrl key while clicking on the server groups then release the Ctrl key.

   b. Click the **Add** button.

   The server groups are added to the list of server groups in the *Servers To Configure Applications On* list box.

   To remove a server group from the *Servers To Configure Applications On* list box, complete the following steps:

   a. Select the server group’s entry in the *Servers To Configure Applications On* list box.
   b. Click the **Remove** button.

   The server group is removed from the list of server groups onto which the selected applications are to be configured.
8. To finish and have the wizard configure the applications onto the server group, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

   If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

**Note:** The wizard does not change the server group until you click the **finish** button on any panel and have specified the details that it needs. This updates the server group within the Configuration specified, but does not change the servers that are members of the server group in your enterprise.

The servers in your enterprise are changed the next time that you activate the Configuration.

**To check that Configuration is valid and create the server group in your enterprise, activate the Configuration.** This automatically creates and starts the application servers on their target server hosts and distributes the applications required to those server hosts.

On the pop-up menu of the Configuration, click **Activate**.

The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

**Note:** If an application needs the Interface Repository (IR) to be populated on target hosts to which it has been distributed by the System Manager, you must do so manually. To update the IR for an application after it has been distributed, complete the following steps on each target host:
1. Copy the IR update utility provided with the application to each target host from the System Manager host
2. Run the IR update utility on the target host

For the name of the IR update utility, and instructions about using it, see the information provided with the application.

### Related Concepts

“Workload Management” on page 335
“Application Servers and Server Groups” on page 25
“Controlled Server Groups” on page 336
“Wizards” on page 447

**Related Tasks**

"Configure Applications onto a Server Group" on page 183
"Configure the Attributes of a Server or Server Group" on page 187
"Configure how Application Servers are to Run" on page 187
"Chapter 5. Configure a new Application Environment" on page 133
"Chapter 7. Administer Application Servers" on page 181

## Configure a new Freestanding Server

Use this procedure to define a new freestanding server that is not part of a server group. Typically, such servers have unique characteristics and support a unique set of applications. This procedure creates a *Server (freestanding)* within a Configuration of one of your application Management Zones and configures the Server (freestanding) onto the Host on which the server is to run. It also configures one or more applications to run on the server.

This task involves the following main stages:

1. Define the Server (free standing) and configure it onto the Host on which the server is to run
2. Configure one or more applications onto the server
3. Activate the Configuration, to check that Configuration is valid and to create the freestanding server on its host. This also automatically copies the application files needed to the server host.

**Considerations:**

- The applications that are to run on the free standing server are configured onto the Server (freestanding). You can do this when configuring a new free standing server as described in this topic, or later as described in "Configure an Application onto a Freestanding Server" on page 186.
- AIX If an application server needs to use shared libraries that are not in one of the directories specified in the LIBPATH, you must make those libraries available before the application server is started. For information about how to do this, see "Make AIX Shared Libraries Available to Application Servers" on page 191.
- If required, after completing the procedure in this topic, you should change the default characteristics of the free standing server. You can do this by editing the attributes of the Server (free standing) and perhaps configuring other system management objects onto it. This is normally dictated by the special requirements of an application, and by detailed information provided with the application family package. Some typical tasks are also provided in related task topics.

**To define the server and configure it onto its host, complete the following steps:**

1. Display the Configuration that the Server (free standing) is to be part of.
2. On the pop-up menu of the Configuration, click **New - Server (free standing)**.

   This displays a dialog box for you to specify a unique name for the new server.
3. Type the name that you want the server to be known by. The name can contain:
   - From 1 through 32 ASCII characters; a through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks

**Note:** The Server (freestanding) cannot have the same name as any other Server (member of group) or any Server Group in the same Configuration.

4. To create the object, click **OK**. If the name is valid, this creates an object in the Servers (free standing) folder. If the name is not valid, you are prompted to enter a new name.

   To cancel the procedure, click **Cancel**.
5. Expand the Servers (free standing) folder. You should see an object with the name specified in step 3. If not, repeat steps 2 to 4.
6. On the pop-up menu of the Server (freestanding), click **Drag**.
7. Display the Host that the server is to run on.
8. On the pop-up menu of the Host, click **Configure server**. This creates a relationship between the object and the Host that it has been configured on.
9. Optionally, to verify that the relationship has been created, expand the Host. You should see a *Configured Host* relationship with the name of the Server (freestanding). If not, repeat steps 6 to 8.

**To configure an application onto the server, complete the following steps:**

1. Expand the Applications folder within the same Configuration as your Server (freestanding)
2. On the pop-up menu of an Application, click **Configure Server**. This creates a *Configured Applications* relationship between the Application and the Server (free standing).

   **Note:** The **Configure Server** action was offered because the Server (freestanding) was still on the system management clipboard from the last **Drag** action.
3. Optionally, to verify that the application has been successfully configured on the server, complete the following steps:
   a. Expand the Server (free standing)
   b. Expand the Configured Applications folder
   c. You should see a *Configured Applications* relationship with the name of the configured Application (from 2). If not, repeat step 2.

**To check that Configuration is valid and create the freestanding server in your enterprise, activate the Configuration.**

On the pop-up menu of the Configuration, click **Activate**.

The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

This task has created a freestanding server with default attribute values. If required, you can edit the Server (freestanding) to specify your own values, as described in later topics.

**Related Concepts**

"Application Servers and Server Groups" on page 25

**Related Tasks**

165
8.h on page 165
“Configure the Attributes of a Server or Server Group” on page 187
“Configure an Application onto a Freestanding Server” on page 186
“Define and Configure Servers and Server Groups” on page 182
“Operate Application Servers” on page 192

## Configure a new Client Style

Use this task to configure a new client style in a Configuration of one of your application Management Zones.

To configure a new client style in your application environment, you can use the **Create Client Style** and **Configure Client Style** wizards, as described in this topic.

This task involves the following main stages:

1. Define the client style and configure it onto one or more hosts
2. Configure any applications needed onto the client style (page 152)
3. Activate the Configuration, to check that Configuration is valid and to create the client style on configured hosts. This also automatically copies the client application files needed to those hosts.

**1. To define the client style, complete the following steps:**

1. On the menu-bar of the System Manager user interface, click **Wizards - Create Client**.
2. On the Management Zone panel, select the management zone within which you want to create the client style. If this panel does not list an appropriate management zone, you can type the name of a new management zone, which will be created automatically by the wizard.
3. Click the **Next** button.
4. On the Configuration panel, select the Configuration (of the management zone that you selected previously) within which you want to create the client style. If this panel does not list an appropriate Configuration, you can type the name of a new Configuration, which will be created automatically by the wizard.
5. Click the **Next** button.
6. On the Client Style panel, specify the parameters needed to create one or more client style. For each client style, specify the following parameters:
   a. The name of the client style. You can either type the name for a new client style or select the name of an existing client style (to configure it onto a new host).
   b. Select the host on which the client style is to be used, from the Host Name pull-down field.
   c. Select bootstrap host for the client style, from the Bootstrap Host Name pull-down field.
7. To finish and have the wizard create the Client Style, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

**Note:** The wizard does not create or configure what you have selected until you click the **finish** button on any panel and have specified the details that it needs. This defines the Client Style within the Configuration specified, but does not add the client style to your enterprise.

The client style created is added to your enterprise the next time that you activate the Configuration.

**2. To configure one or more applications onto the client style, complete the following steps:**

1. On the menu-bar of the System Manager user interface, click **Wizards - Configure Client**.
2. On the Select Applications to Configure panel, select the applications that you want to configure onto the client style.

   To add an application to the list to be configured, displayed under *Applications to Configure*, complete the following steps:

   a. In the *Available Applications* list box, click on one or more applications. To select several applications, press and hold the Ctrl key while clicking on the applications then release the Ctrl key.

   b. Click the **Add** button.

   The applications are added to the list of applications to be configured on the client style, listed in the *Applications to Configure* list box.

   To remove an application from the *Applications to Configure* list box, complete the following steps:

   a. Select the application's entry in the *Applications to Configure* list box.

   b. Click the **Remove** button.

   The application is removed from the list of applications to be configured on the client style.
3. Click the **Next** button.
4. On the Management Zone panel, select the application Management Zone within which the client style is defined.
5. Click the **Next** button.
6. On the Configuration panel, select the Configuration (of the Management Zone that you selected previously) within which the client style is defined.
7. Click the **Next** button.
8. On the Client Style panel, select the name of one or more client styles onto which the applications are to be configured.

   To add a client style to the list under *Client Styles To Configure Applications On*, complete the following steps:

   a. In the *Available Client Styles* list box, click on one or more client styles. To select several client styles, press and hold the Ctrl key while clicking on the client styles then release the Ctrl key.

   b. Click the **Add** button.

The client styles are added to the list of client styles in the *Client Styles To Configure Applications On* list box.

To remove a client style from the *Client Styles To Configure Applications On* list box, complete the following steps:

a. Select the client style's entry in the *Client Styles To Configure Applications On* list box.
b. Click the **Remove** button.

The client style is removed from the list of client styles onto which the selected applications are to be configured.

9. To finish and have the wizard configure the applications onto the client style, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

   If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

**Note:** The wizard does not change the client style until you click the **finish** button on any panel and have specified the details that it needs. This updates the client style within the Configuration specified, but does not change the clients that use that client style in your enterprise.

The clients in your enterprise are changed the next time that you activate the Configuration.

**3. To check that Configuration is valid and create the client style in your enterprise, activate the Configuration.**

On the pop-up menu of the Configuration, click **Activate**.

The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

**Note:** If you set an application server's **run control** attribute to `run on request`, then you should **"Configure the Request Timeout for Client Styles" on page 214** attribute for all clients of that server. The default request time out value for client styles is too short when an application server's **run control** attribute is set to `run on request`. This is because a run on request application server must be started when it receives its first request. The response time for this first request is therefore much longer than the response time for subsequent requests.

**Related Concepts**

"Client Styles" on page 26
"Clients" on page 207
"Wizards" on page 447

**Related Tasks**

"Configure Applications onto a Client Style" on page 213
"Configure the Request Timeout for Client Styles" on page 214

## Example: Configure a new Single-host Application Environment

This topic describes how to configure a Component Broker application environment to run on a standalone host, based on the example shown in the figure Example: The Single-host Application Environment (page Figure 38 on page 155). The tasks to be completed configure the applications, application servers, and client styles onto the host that they are to run on.

It provides an example of the general procedure described in "Chapter 5. Configure a new Application Environment" on page 133 applied to a standalone host.

This example makes the following assumptions:

- You have already activated a system management Configuration that defines the cell, work group, and serving host of the Component Broker host environment. If you have not done this, see "Example: Configure a new Single-host Environment" on page 91.
- You have already compiled the sample Policy application provided with Component Broker. This example makes use of the sample Policy application; if you want to use that application, you should have compiled it as described in ″Compiling the Sample Application″ in the *Quick Beginnings*.

## An Overview of the Example Application Environment

The example application environment (shown in Figure 38 on page 155) comprises the following components on the standalone host *host1.cbnet.com*.

*Table 12.* **Components to Configure for the Example Single-host Application Environment**

| Component | Name | Your value? | Applications Run |
|---|---|---|---|
| **Server Group** | Test Server Group | | Policy ... |
| **Servers (member of group)** | Test Server 1<br>Test Server 2 | | n/a (defined on Server Group) |
| **Client Style** | Test Client Style | | |
| **Management Zone** | Application Test Zone | | n/a |
| **Configuration** | Application Test Configuration | | n/a |

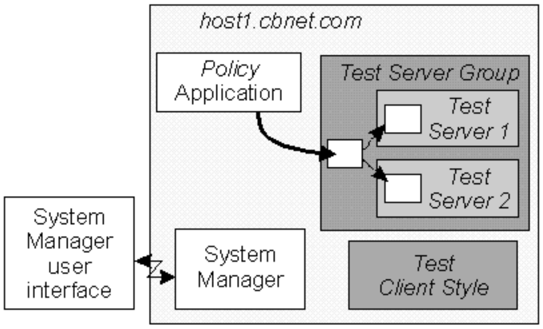


*Figure 38.* **The Example Application Environment**

## The Task to Configure the Example Single-Host Application Environment

Use this task to configure the example single-host environment shown in Figure 38. The task creates and configures the system management objects listed in the table Components of the Example Application Environment (page 154). If you want to use the example to configure your own application environment, substitute the names listed with your own names.

The task involves the following subtasks, described below:

1. Create a new Management Zone to be used for your application environment
2. Create a Configuration within your Management Zone
3. Load the Policy application into your application environment
4. Add the new application into the Configuration
5. Define the server group and the servers that are members of the server group
6. Configure the *Policy* application onto the server group
7. Define the application client style
8. Activate the Configuration.

**Note:** The system management wizards used do not create or configure what you have selected until you click the **finish** button on any panel and have specified the details that it needs. The wizards define the system management objects needed within the Configuration specified, but do not change your enterprise. Your enterprise is updated the next time that you activate the Configuration.

**To configure the example single-host application environment, complete the following steps:**

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. **Create a new Management Zone to be used for your application environment**.

   You are recommended to use application Management Zones only for configuring your application environment, and use a separate Management Zone for configuring your host network.

   To create a new Management Zone, complete the following steps:

   a. On the pop-up menu of the Management Zones folder, click **Insert**

    This opens a dialog box for you to specify a unique name for the new Management Zone.

   b. Type the name *Application Test Zone*
   c. To create the new Management Zone, click the **OK** button.
   d. Expand the Management Zones folder, by clicking its + sign. You should see an object for the Management Zone that you created. If not, repeat the previous steps.

3. **Create a Configuration within your Management Zone**.

   Each Management Zone must contain at least one Configuration, which defines an implementation of the Management Zone.

   a. On the pop-up menu of *Application Test Zone*, click **New - Configuration**.

      This opens a dialog box for you to specify a unique name for the new Configuration.

   b. Type the name *Application Test Configuration*
   c. To create the new Configuration, click the **OK** button.
   d. Expand the Configurations folder. You should see a Configuration called *Application Test Configuration*. If not, repeat the previous steps.

4. **Load the Policy application into your application environment**.

   This step copies the application files to a unique directory under Component Broker and creates the system management objects used to represent and manage the application. It also adds the application to the *Application Test Configuration*. This enables you to later configure the application onto servers or client styles and enables the System Manager to automatically copy application files to where they are needed.

   Normally, this procedure is completed automatically by the application installation tool. However, this example uses the **Load Application** action through the System Manager user interface.

   To load the Policy application, complete the followng steps:

   a. Check that you are in *Expert user* mode; for example, click on the *Expert user* icon of the tool bar. (To use the **Load Application** action, you need to be in *Expert user* mode.)
   b. Expand the Host Images folder, to display the icons for managed hosts.
   c. From the pop-up menu of the System Manager's host, click **Load Application**.

      (If you are unsure which icon is for the System Manager's host, the name of the System Manager is displayed on the title bar of the Information Controller window.)

   d. In the entry field of the *Load Application* dialog window, type the full path name of the Policy application's DDL file.

      You can use the **Browse** button to search for and select the application's DDL files.

      - WIN

        ```
        x:\CBroker\samples\InstallVerification\ProgrammingModel\BusinessObjects
              \Policy\Working\NT\PolicyFamily\PolicyFamily.ddl
        ```

        where `x:\CBroker` represents the installation directory for Component Broker.

      - AIX

```
/cbfs/cbuser/samples/InstallVerification/ProgrammingModel/BusinessObjects
      /Policy/Working/AIX/PolicyFamily/PolicyFamily.ddl
```

where `/cbfs/cbuser` is the $HOME directory of the Component Broker user.

**Notes:**

1) The Browse dialog box lists directories visible on the workstation ( WIN ) or host ( AIX ) on which the System Manager user interface is running, regardless of which Host Image you select the action on.
2) ( WIN ) To load a DDL file on a remote host (for example, if the System Manager is running on a remote host), you can map the network drive into which the application was installed on that remote host. Make sure that the account you use to map the network drive has the same permissions as the account under which you are running the CBConnector service.

e. An Action Console window is displayed. When the action has completed, the window contains a message indicating successful completion.

f. Close the Action Console window.

5. **To add the new application into** *Application Test Configuration***, complete the following steps:**

   a. Scroll to the top of the View panel

   b. Expand the Available Applications folder You should see an Application with the name *Policy* that you just loaded.

   c. On the pop-up menu of the *Policy* Application, click **Drag**

   d. Scroll down to *Application Test Configuration*

   e. On the pop-up menu of *Application Test Configuration*, click **Add Application**

      This automatically creates the Application with the attributes and relationships that it needs.

   f. Expand the Configuration *Application Test Configuration*.

   g. Expand the Applications folder. You should see an Application called *Policy*.

6. **To define the server group and the servers that are members of the server group, complete the following steps:**

   a. On the menu-bar of the System Manager user interface, click **Wizards - Create Servers**

   b. On the Management Zone panel, select *Application Test Zone*.

   c. Click the **Next** button.

   d. On the Configuration panel, select *Application Test Zone*, within which you want to create the server group.

   e. Click the **Next** button.

   f. On the Server Group panel, type *Test Server Group*.

   g. Click the **Next** button.

   h. On the Servers panel, add the servers to the group, displayed in the list box, by completing the following steps:

      1) Type *Test Server 1*
      2) Select the host, *host1.cbnet.com*, on which the server is to run.
      3) Click the **add** button

4) Type *Test Server 2*

5) Select the host, *host1.cbnet.com*, on which the server is to run.

6) Click the **add** button

Each server, and its host, is added to the server group's list box.

i. To finish and have the Create Server wizard create the server group, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

   If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

**Note:** The wizard does not create or configure what you have selected until you click the **finish** button on any panel and have specified the details that it needs. This defines the server group within the Configuration specified, but does not add the servers to your enterprise.

The server group is added to your application environment the next time that you activate the Configuration.

7. **To configure the** *Policy* **application onto the server group, complete the following steps:**
   a. On the menu-bar of the System Manager user interface, click **Wizards - Configure Servers**.
   b. On the Select Applications to Configure panel, click *Policy*.
   c. Click the **Add** button.

      The application is added to the list of applications to be configured on the server group, listed in the *Applications to Configure* list box.
   d. Click the **Next** button.
   e. On the Management Zone panel, select *Application Test Zone*.
   f. Click the Next button.
   g. On the Configuration panel, select *Application Test Configuration*.
   h. Click the **Next** button.
   i. On the Server Group panel, select *Test Server Group*.
   j. Click the **Add** button.

      The server group is added to the list of server groups in the *Servers To Configure Applications On* list box.
   k. To finish and have the wizard configure the application onto the server group, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

      If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

**Note:** The wizard does not change the server group until you click the **finish** button on any panel and have specified the details that it needs. This updates the server group within the Configuration specified, but does not change the servers that are members of the server group in your enterprise.

The servers in your enterprise are changed the next time that you activate the Configuration.

8. **To define the application client style, complete the following steps:**
   a. On the menu-bar of the System Manager user interface, click **Wizards - Create Client**.
   b. On the Management Zone panel, select *Application Test Zone*.
   c. Click the **Next** button.
   d. On the Configuration panel, select *Application Test Configuration*.
   e. Click the **Next** button.
   f. On the Client Style panel, specify the following parameters needed to create the client style:
      - **Name** *Test Client Style*
      - **Host Name** *host1.cbnet.com*, from the Host Name pull-down field.
      - **Bootstrap Host Name** *host1.cbnet.com*, from the Bootstrap Host Name pull-down field.
   g. Click the **Add** button.
   h. To finish and have the wizard create the Client Style, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

      If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.
9. **To check that Configuration is valid and create the server group in your enterprise, activate the Configuration.**

   On the pop-up menu of the Configuration, click **Activate**.

   The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

You have now configured the single-host application environment shown in Figure 38 on page 155.

## Example: Configure a new Multi-host Application Environment

This topic describes how to configure a Component Broker application environment to run in a multi-host environment, based on the example shown in Figure 39 on page 161. The tasks to be completed configure the applications, application servers, and client styles onto the hosts that they are to run on.

It provides an example of the general procedure described in “Chapter 5. Configure a new Application Environment” on page 133 applied to a multi-host environment.

This example makes the following assumptions:

- You have already activated a system management Configuration that defines the cell, work group, and serving host of the Component Broker host environment. If you have not done this, see “Example: Configure a new Multi-Host Network” on page 121.
- You have already compiled the applications to be configured and installed them into the appropriate application directory on the System Manager host. For example, WIN *c:\Cbroker\NtApps\LifeInsFamily*. If you want to use the sample Policy application provided with Component Broker, you should have compiled it as described in ″Compiling the Sample Application″ in the *Quick Beginnings*.

## An Overview of the Example Application Environment

The example application environment (shown in Figure 39 on page 161) comprises the following components:

*Table 13. Components to Configure for the Example Multi-host Application Environment*

| **Component** | **Name** | **Your value?** | **Applications Run** |
|---|---|---|---|
| **Server Group** | LifeIns Server Group | | LifeIns ... |
| **Servers (member of group)** | LifeIns Server 21<br>LifeIns Server 31 | | n/a (defined on Server Group) |
| **Server (freestanding)** | Accounts Server 21 | host2.cbnet.com | Accounts ... |
| **Client Style (Java)** | javains Client Style | host2.cbnet.com | |
| **Client Style (Managed)** | plmaview Client Style | host3.cbnet.com<br>host4.cbnet.com | |
| **Management Zone** | LifeIns Application Zone | | n/a |
| **Configuration** | LifeIns Application Configuration | | n/a |

*Table 14.* **Example Applications**

| **Application** | **DDL Files** | **Your application?** | **Your DDL files?** |
|---|---|---|---|
| **LifeIns** | BaseLifeIns.ddl<br>SpecificLifeIns.ddl | | |
| **Accounts** | AccountFamily.ddl | | |

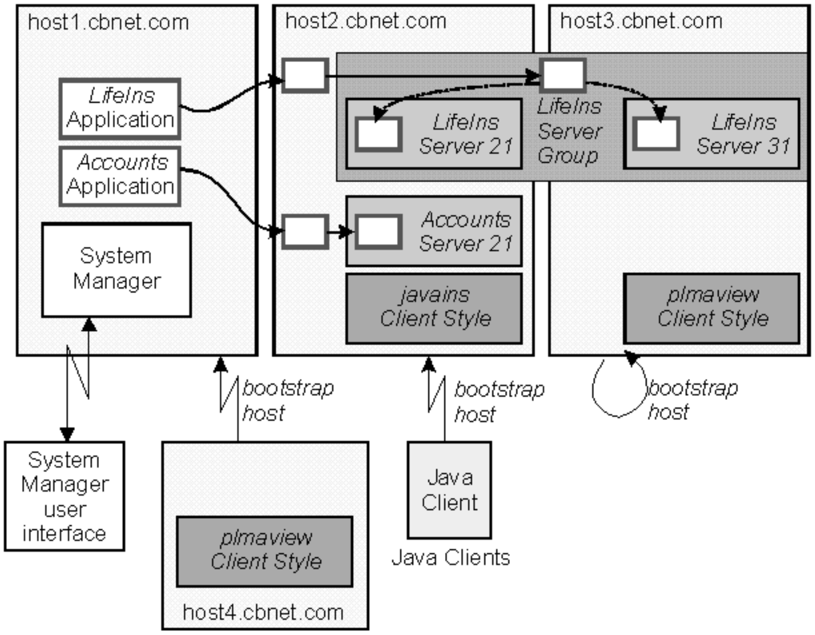


*Figure 39. The Example Application Environment*

## The Task to Configure the Example Multi-Host Application Environment

Use this task to configure the example multi-host environment shown in Figure 39. The task creates and configures the system management objects listed in Table 13 on page 160. If you want to use the example to configure your own application environment, substitute the names listed with your own names.

The task involves the following subtasks, described below:

1. Create a new Management Zone to be used for your application environment
2. Create a Configuration within your Management Zone
3. Load the applications into your application environment
4. Add the new applications into the Configuration
5. Define the server group and the servers that are members of the server group
6. Configure the application onto the server group
7. Define the freestanding server
8. Configure the application onto the freestanding server
9. Define the application client styles
10. Activate the Configuration.

**Note:** The system management wizards used do not create or configure what you have selected until you click the **finish** button on any panel and have specified the details that it needs. The wizards define the system management objects needed within the Configuration specified, but do not change your enterprise. Your enterprise is updated the next time that you activate the Configuration.

**To configure the example multi-host application environment, complete the following steps:**

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. **Create a new Management Zone to be used for your application environment**.

   You are recommended to use application Management Zones only for configuring your application environment, and use a separate Management Zone for configuring your host network.

   To create a new Management Zone, complete the following steps:

   a. On the pop-up menu of the Management Zones folder, click **Insert**

      This opens a dialog box for you to specify a unique name for the new Management Zone.

   b. Type the name *LifeIns Application Zone*

   c. To create the new Management Zone, click the **OK** button.

   d. Expand the Management Zones folder, by clicking its + sign. You should see an object for the Management Zone that you created. If not, repeat the previous steps.
3. **Create a Configuration within your Management Zone**.

   Each Management Zone must contain at least one Configuration, which defines an implementation of the Management Zone.

   a. On the pop-up menu of *LifeIns Application Zone*, click **New - Configuration**.

      This opens a dialog box for you to specify a unique name for the new Configuration.

   b. Type the name *LifeIns Application Configuration*

   c. To create the new Configuration, click the **OK** button.

   d. Expand the Configurations folder. You should see a Configuration called *LifeIns Application Configuration*. If not, repeat the previous steps.
4. **Load the applications into your application environment**.

   This step copies the application files to a unique directory under Component Broker and creates the system management objects used to represent and manage the application. It also adds the application to the *LifeIns Application Configuration*. This enables you to later configure the application onto servers or client styles and enables the System Manager to automatically copy application files to where they are needed.

   Normally, this procedure is completed automatically by the application installation tool. However, this example uses the **Load Application** action through the System Manager user interface.

   To load the applications, complete the following steps:

   a. Check that you are in *Expert user* mode; for example, click on the *Expert user* icon of the tool bar. (To use the **Load Application** action, you need to be in *Expert user* mode.)

   b. Expand the Host Images folder, to display the icons for managed hosts.

   c. From the pop-up menu of the System Manager's host, click **Load Application**.

      (If you are unsure which icon is for the System Manager's host, the name of the System Manager is displayed on the title bar of the Information Controller window.)

   d. In the entry field of the *Load Application* dialog window, type the full path name of the application's DDL file.

   You can use the **Browse** button to search for and select the application's DDL files.

   e. An Action Console window is displayed. When the action has completed, the window contains a message indicating successful completion.
   f. Close the Action Console window.

   **Notes:**

   a. Repeat this step for each of the application DDL files (for this example, listed in the table Example Applications (page Figure 39 on page 161).
   b. The *LifeIns* application uses queriable objects, so has a second *specific*family.ddl* file, which has the information needed to do queries. Loading the *specificLifeIns.ddl* file, adds the extra information to the system management objects created for the application.
   c. The Browse dialog box lists directories visible on the workstation ( WIN ) or host ( AIX ) on which the System Manager user interface is running, regardless of which Host Image you select the action on.
   d. ( WIN ) To load a DDL file on a remote host (for example, if the System Manager is running on a remote host), you can map the network drive into which the application was installed on that remote host. Make sure that the account you use to map the network drive has the same permissions as the account under which you are running the CBConnector service.

5. **To add the new applications into *LifeIns Application Configuration*, complete the following steps:**
   a. Scroll to the top of the View panel
   b. Expand the Available Applications folder You should see Application objects called *LifeIns* and *Accounts*.
   c. On the pop-up menu of *LifeIns*, click **Drag**
   d. Scroll down to *LifeIns Application Configuration*
   e. On the pop-up menu of *LifeIns Application Configuration*, click **Add Application**

      This automatically creates the Application with the attributes and relationships that it needs.
   f. Expand the Configuration *LifeIns Application Configuration*.
   g. Expand the Applications folder. You should see an Application called *LifeIns*.

   Repeat this step for the *Accounts* Application.

6. **To define the server group and the servers that are members of the server group, complete the following steps:**
   a. On the menu-bar of the System Manager user interface, click **Wizards - Create Servers**
   b. On the Management Zone panel, select *LifeIns Application Zone*.
   c. Click the **Next** button.
   d. On the Configuration panel, select *LifeIns Application Zone*, within which you want to create the server group.
   e. Click the **Next** button.
   f. On the Server Group panel, type *LifeIns Server Group*.
   g. Click the **Next** button.

h. On the Servers panel, add the servers to the group, displayed in the list box, by completing the following steps:

   1) Type *LifeIns Server 21*
   2) Select the host, *host2.cbnet.com*, on which the server is to run.
   3) Click the **add** button
   4) Type *LifeIns Server 31*
   5) Select the host, *host3.cbnet.com*, on which the server is to run.
   6) Click the **add** button

   Each server, and its host, is added to the server group's list box.

i. To finish and have the Create Server wizard create the server group, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

   If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

**Note:** The wizard does not create or configure what you have selected until you click the **finish** button on any panel and have specified the details that it needs. This defines the server group within the Configuration specified, but does not add the servers to your enterprise.

The server group is added to your application environment the next time that you activate the Configuration.

7. **To configure the *LifeIns* application onto the server group, complete the following steps:**

   a. On the menu-bar of the System Manager user interface, click **Wizards - Configure Servers**.
   b. On the Select Applications to Configure panel, click *LifeIns*.
   c. Click the **Add** button.

      The application is added to the list of applications to be configured on the server group, listed in the *Applications to Configure* list box.
   d. Click the **Next** button.
   e. On the Management Zone panel, select *LifeIns Application Zone*.
   f. Click the Next button.
   g. On the Configuration panel, select *LifeIns Application Configuration*.
   h. Click the **Next** button.
   i. On the Server Group panel, select *LifeIns Server Group*.
   j. Click the **Add** button.

      The server group is added to the list of server groups in the *Servers To Configure Applications On* list box.
   k. To finish and have the wizard configure the application onto the server group, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

      If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

**Note:** The wizard does not change the server group until you click the **finish** button on any panel and have specified the details that it needs. This updates the server group within the Configuration specified, but does not change the servers that are members of the server group in your enterprise.

The servers in your enterprise are changed the next time that you activate the Configuration.

8. **To define the freestanding server, complete the following steps:**
   a. On the pop-up menu of the Configuration *LifeIns Configuration*, click **New - Server (free standing)**.

      This displays a dialog box for you to specify a unique name for the new server.

   b. Type the name *Accounts Server 21*.
   c. To create the object, click **OK**.
   d. Expand the Servers (free standing) folder. You should see a Server (freestanding) called *Accounts Server 21*. If not, repeat steps i to iii.
   e. On the pop-up menu of the Server (freestanding), click **Drag**.
   f. Display the Host, *host2.cbnet.com*, that the server is to run on.
   g. On the pop-up menu of *host2.cbnet.com*, click **Configure server**. This creates a relationship between *Accounts Server 21* and *host2.cbnet.com*.
   h. Optionally, to verify that the relationship has been created, expand *Accounts Server 21*. You should see a *Configured Host* relationship named *host2.cbnet.com*. If not, repeat steps v to vii.
9. **To configure the** *Accounts* **application onto the freestanding server** *Accounts Server 21***, complete the following steps:**
   a. Expand the Applications folder within the Configuration *LifeIns Application Configuration*.
   b. On the pop-up menu of the *Accounts* Application, click **Configure Server**. This creates a *Configured Applications* relationship between the Application and the Server (free standing).

      **Note:** The **Configure Server** action was offered, because *Accounts Server 21* was still on the system management clipboard from the last **Drag** action.

   c. Optionally, to verify that the application has been successfully configured on the server, complete the following steps:
      1) Expand *Accounts Server 21*
      2) Expand the Configured Applications folder
      3) You should see a *Configured Applications* relationship with the name *Accounts*. If not, repeat step ii.
10. **To define the application client styles, complete the following steps:**
    a. On the menu-bar of the System Manager user interface, click **Wizards - Create Client**.
    b. On the Management Zone panel, select *LifeIns Application Zone*.
    c. Click the **Next** button.
    d. On the Configuration panel, select *LifeIns Application Configuration*.
    e. Click the **Next** button.
    f. On the Client Style panel, specify the following parameters needed to create the *javains* client style:
       - **Name** *javains Client Style*
       - **Host Name** *host2.cbnet.com*, from the Host Name pull-down field.

- **Bootstrap Host Name** *host2.cbnet.com*, from the Bootstrap Host Name pull-down field.

g. Click the **Add** button.

h. On the Client Style panel, specify the following parameters needed to create the *plmaview* client style for *host3.cbnet.com*:

- **Name** *plmaview Client Style*
- **Host Name** *host3.cbnet.com*, from the Host Name pull-down field.
- **Bootstrap Host Name** *host3.cbnet.com*, from the Bootstrap Host Name pull-down field.

i. Click the **Add** button.

j. On the Client Style panel, specify the following parameters needed to create the *plmaview* client style for *host4.cbnet.com*:

- **Name** *plmaview Client Style*
- **Host Name** *host4.cbnet.com*, from the Host Name pull-down field.
- **Bootstrap Host Name** *host1.cbnet.com*, from the Bootstrap Host Name pull-down field.

k. Click the **Add** button.

l. To finish and have the wizard create the Client Style, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

11. **To check that Configuration is valid and create the server group in your enterprise, activate the Configuration.**

On the pop-up menu of the Configuration, click **Activate**.

The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

You have now configured the application environment shown in Figure 39 on page 161.

When you activate a Configuration containing a Java Client Style, the System Manager creates the Client Style Image on the hosts on which the Client Style is configured, and produces the corresponding client styles properties file at each of those hosts. It is then up to you to copy the client style properties file to somewhere that your Java clients can locate them.

With either a Java application client or Java applet client you can place the client style properties file on a Web Server; although this is much more prevalent for Java applet clients. When the client style properties are placed on a Web Server, they can be shared by many clients. For those clients that you want to share the same properties file, you only need to specify the same URL when the client is invoked.

# Chapter 6. Administer your Host Environment

This topic provides information about administering hosts in an existing Component Broker host environment.

If you have not yet configured your host network, you should see one of the following topics:

- “Chapter 3. Create a New Single-Host Enterprise” on page 85
- “Configure a Multi-Host Environment” on page 109

If you want to administer the application environment running on your Component Broker host network, you should see the related topics given at the end of this topic.

The tasks described in this topic can be used whenever required, and are not given in any chronological order. Any changes that you make to the configuration of your host network must conform to the “Name Tree Configuration Management Rules” on page 168.

The following tasks can be used to administer your host network:

- “Verify that System Management Components are Running” on page 169
- “Configure a new Work Group” on page 115
- “Configure a new Host into your Network” on page 117
- “Verify that a Name Server is in the DCE CDS” on page 119
- “Configure a Host as a Member of Non-preferred Work Groups” on page 118
- “Define a TCP/IP or IPC Protocol” on page 170
- “Define a Name Server” on page 171
- “Configure a Remote Name Context for Use on Hosts” on page 172
- “Change the DCE/CDS Confidence Level” on page 175
- “Configure Security for a Host Daemon” on page 329
- “Remove a Name Server from a Host” on page 176
- “Uninstall Managed Hosts” on page 178

**Related Concepts**

“Administration of Enterprises” on page 6
“Hosts” on page 20
“Cells and Work Groups” on page 19
“The System Name Tree” on page 21
“Remote Name Context Connections to other System Name Trees” on page 23
“The Component Broker Name Tree and DCE” on page 23

**Related Tasks**

“Chapter 7. Administer Application Servers” on page 181
“Chapter 8. Administer Clients” on page 205
“Load a new Application” on page 141
“Chapter 10. Administer Management Zones and Configurations” on page 253
“Chapter 11. Administer Security in your Enterprise” on page 261

“Chapter 13. Administer Connections to Tier-3 Systems” on page 343
“Example: Configure a new Multi-Host Network” on page 121

## Name Tree Configuration Management Rules

This topic contains the rules that govern how the name tree is configured and the allowable changes that can be made. Not all of these rules are enforced by the System Manager. Therefore, these rules must be understood and used correctly. When these rules are not followed, any number of different failures with the run-time environment can occur.

The following rules overlap, but are provided for clarity:

- The Component Broker network can contain only one cell.
  - Each host in the Component Broker network must prefer the one cell and must be a member of it.
  - The host a cell prefers cannot be changed, because only one cell can be in the Component Broker network.
  - The only time the cell can be removed is when the host that serves the cell is the only host remaining in the Component Broker network and all references to that host are being removed from the configuration in preparation for the Name Server removal.
- The Component Broker network must contain at least one workgroup.
  - The workgroup a host prefers cannot be changed.
  - A new workgroup can be added at any time, provided its serving host already serves at least one other workgroup.
  - A workgroup can be removed at any time, provided there are no hosts that prefer that workgroup.
  - The only time the last workgroup on a host (the workgroup preferred by that host) can be removed is when all references to that host are being removed from the configuration in preparation for the Name Server removal.
- Each host in the Component Broker network must prefer one workgroup and must be a member of that preferred workgroup.
  - If a host serves one or more workgroups, that host must serve the workgroup that it prefers.
  - A host can be a member of any number of other non-preferred workgroups.
- In a Component Broker network, there is one host that serves the cell and serves a workgroup. That workgroup is the minimum required workgroup for a Component Broker network.
  - Any other host in the Component Broker network can serve from zero-to-any number of workgroups.
  - Whether a host serves zero workgroups or 1-to-n workgroups is determined the first time that a Configuration containing that host is activated. That is, if a host is to ever serve any workgroup, it must have its ″served workgroups″ relationship created with a workgroup before the first time that its Configuration is activated. A host can never switch between serving and not serving workgroups.
- The only time a Name server should be removed is in preparation for the uninstall of that host.

**Related Concepts**

"The System Name Tree" on page 21
"The Component Broker Name Tree and DCE" on page 23
"Cells and Work Groups" on page 19
"Management Zones" on page 17
"Hosts" on page 20

**Related Tasks**

"Create a new Management Zone and Configuration for your Host Network" on page 110
"Configure the Component Broker Cell and Minimum Work Group" on page 111
"Configure a new Work Group" on page 115
"Configure a new Host into your Network" on page 117
"Verify that a Name Server is in the DCE CDS" on page 119
"Chapter 6. Administer your Host Environment" on page 167

## Verify that System Management Components are Running

Use this procedure to verify that CBConnector System Management components are running.

WIN The System Manager and Agent run as the Windows NT service called CBConnector. An initial check that the service is running is to display the Windows NT Services window.

AIX The System Manager and Agent run as the AIX process called **bgmain**. An initial check that the process is running is to list all active processes and look for that name.

**System Manager**

Start a System Manager user interface that is configured to use the System Manager. If the user interface is displayed, the System Manager is running. Otherwise, respond to the error messages displayed; for example, to check that the CBConnector service is started.

**SM agent**

Use the CBConnector System Management user interface to display Image objects for the Host Image that the agent is managing. To see the Host Images folder, you need to be in *Expert user* mode; for example, click on the *Expert user* icon of the tool bar.

If the state of the Image objects is displayed on the Status bar at the bottom of the Information Controller window, then the agent is running. Otherwise, check that the CBConnector service is started on that host.

**Other components**

For other components, such as servers and daemons, use the CBConnector System Management user interface to display and act on the objects representing components. When you click on a component object, its state is displayed on the Status bar at the bottom of the Information Controller window. You can use actions specific to the object type to change the state of the component; for example, to start or stop a server.

**Related Concepts**

"System Manager User Interface" on page 12

"Appendix A. Features of the User Interface" on page 433
"Chapter 2. Use the System Manager User Interface" on page 55

## Add a new Configuration to the Management Zone for your Host Network

Complete the steps in this topic to create a new Configuration within the Management Zone used to configure your host network.

You create the Management Zone and its initial Configuration when you first configure your host network, as described in "Create a new Management Zone and Configuration for your Host Network" on page 110.

The Configuration created in this topic is to provide an alternative configuration of hosts within your Component Broker network. However, the contents of the new Configuration must still conform to the "Name Tree Configuration Management Rules" on page 168.

**To add a new Configuration to the Management Zone for your Host Network, complete the following steps:**

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Expand the Management Zones folder
3. On the pop-up menu of your Management Zone used for your host network, click **New - Configuration**.

   This opens a dialog box for you to specify a unique name for the new Configuration. Type the name that you want the new object to be known by. The name can contain:

   - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks
4. To create the new Configuration, click the **OK** button. If you specified a valid name, this creates a new Configuration. If the name specified is not valid, you are prompted to enter a new name.
5. Expand the Configurations folder of your Management Zone. You should see an object for the Configuration that you created. If not, repeat the previous steps.

Next "Configure the Component Broker Cell and Minimum Work Group" on page 111, including the host that serves both the cell and minimum work group.

## Define a TCP/IP or IPC Protocol

Use this task to define a model TCP/IP or IPC protocol that you can configure onto one or more hosts that need such a protocol with the same characteristics.

The task creates either a *TCP/IP Protocol* or *IPC Protocol* system management object within the Configuration of the Management Zone that you are using for your Component Broker host network. You can configure the TCP/IP Protocol or IPC Protocol onto one or more hosts in that Configuration, to create protocols with the same characteristics. This task ends when you have defined the TCP/IP Protocol or IPC Protocol.

**Prerequisites:**

This topic assumes that you have created a Management Zone and Configuration to be used to configure your Component Broker network, as described in “Create a new Management Zone and Configuration for your Host Network” on page 110.

**To define a TCP/IP protocol, complete the following steps:**

1. Display the Configuration of the Management Zone that you are using to configure your host network.
2. On the pop-up menu of the Configuration, click either **New - TCP/IP Protocol** or **New - IPC Protocol**.

   This displays a dialog box for you to specify a unique name for the new protocol.
3. Type the name that you want the protocol to be known by. The name can contain:
   - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks

   This name is used only to identify the TCP/IP Protocol or IPC Protocol on the System Manager user interface.
4. To create the object, click the **OK** button. If the name is valid, this creates the system management Protocol object. If the name is not valid, you are prompted to enter a new value.

   To cancel the procedure, click the **Cancel** button.
5. Optionally, to verify that the object has been created, expand either the TCP/IP Protocols folder or the IPC Protocols folder, as appropriate. You should see an object with the name specified in step 3. If not, repeat steps 2 to 4.

After you have defined a new TCP/IP Protocol, you can use it when adding a new host to your Component Broker host network, as described in “Configure a new Host into your Network” on page 117.

Otherwise, you can configure the TCP/IP Protocol or IPC Protocol onto an existing Host, as described in “Configure a TCP/IP or IPC Protocol onto a Host” on page 174.

**Related Tasks**

“Create a new Management Zone and Configuration for your Host Network” on page 110
“Configure a new Host into your Network” on page 117

## Define a Name Server

Use this task to define a model name server that you can configure onto one or more hosts that need a host name server with the same characteristics.

The task creates a Name Server system management object within the Configuration of the Management Zone that you are using for your Component Broker host network. You can configure the Name Server onto one or more hosts in that Configuration, to create host name servers with the same characteristics. This task ends when you have defined the Name Server.

**Prerequisites:**

This topic assumes that you have created a Management Zone and Configuration to be used to configure your Component Broker network, as described in "Create a new Management Zone and Configuration for your Host Network" on page 110.

**To define a Name server, complete the following steps:**

1. Display the Configuration of the Management Zone that you are using to configure your host network.
2. On the pop-up menu of the Configuration, click **New - Name Server**

   This displays a dialog box for you to specify a unique name for the new Name Server.
3. Type the name that you want the Name Server to be known by. The name can contain:
   - From 1 through 32 ASCII characters; a through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks
4. To create the Name Server, click **OK**. If the name is valid, this creates an object in the Name Servers folder. If the name is not valid, you are prompted to enter a new name.

   To cancel the procedure, click **Cancel**.
5. Optionally, to verify that the object has been created, expand the Name Servers folder. You should see an object with the name specified in step 3. If not, repeat steps 2 to 4.

After you have defined a new Name Server, you can use it when adding a new host to your Component Broker host network, as described in "Configure a new Host into your Network" on page 117.

By default, the name server reads data from the most convenient DCE/CDS server. However, you can configure the name server to get data from the CDS clerk's cache or from the server where the master replica is located, depending on the degree to which the name server can trust the accuracy of data returned from the DCE/CDS server. You do this by "Change the DCE/CDS Confidence Level" on page 175.

**Related Concepts**

"The System Name Tree" on page 21
"Name Tree Configuration Management Rules" on page 168

**Related Tasks**

"Create a new Management Zone and Configuration for your Host Network" on page 110
"Configure a new Host into your Network" on page 117
"Change the DCE/CDS Confidence Level" on page 175

## Configure a Remote Name Context for Use on Hosts

Use this procedure to define "Remote Name Context Connections to other System Name Trees" on page 23 bindings to be used to link the name tree for a host to a remote name tree. Each remote name context binding is defined by a *Remote Name Context* model in your Configuration. When you next activate your

Configuration after completing this procedure, the remote name context bindings identified in the Remote Name Context objects will be bound into the name tree used by the host according to their attribute settings.

**To define remote name context bindings to be used by a host, complete the following steps:**

1. Display the Configuration within which you want to define the Remote Name Context models.
2. Create a Remote Name Context model, to define the remote name context bindings. To do this, on the pop-up menu of the Configuration, click **New - Remote Name Context**.
3. Specify details of the remote name context bindings by editing the attributes of the Remote Name Context model. To do this, complete the following steps:
    a. On the pop-up menu of the Remote Name Context model, click **Edit**, to display the Object Editor window.
    b. In the Object Editor window, click the Main tab.
    c. Define the scope of the location, by changing appropriate **scope** attributes from the following list:

**remote host modifier**
The transport modifier for the remote host in the form *iiop://host.ip.addr:port*; where *host.ip.addr* specifies the IP address of the remote host, and *port* specifies the numeric IP port that the remote host expects to use.

**remote name context**
The string form of the name needed to resolve to the remote name context from the local root of the remote host; for example, `.:/workgroups/cbworkstations/hosts`, where the name tree for the local host is to be bound as a member host into the work group named **cbworkstations** on the remote host.

**local name context**
The string form of the name needed to resolve to the local name context from the local root; for example, `host`, where the host name tree for the local host is to be bound as a member *host* into the remote name context on the remote host.

**binding direction**
Whether the local name context is bound into the remote name context, or the other way around; default, **`local into remote`**.

**`local into remote`**
The local name context is bound into the remote name context. This option should not be used when the remote name context is part of a Component Broker Windows NT or AIX host environment. In this case, you need to define the relationship in the other network's System Manager and use **remote into local**.

**`remote into local`**
The remote name context is bound into the local name context

**binding name**
The name used to bind one name context into the other; for example,

**binding direction** = `local into remote`
**binding name** = `cbnt.austin.ibm.com`

Where the host name tree for the host called *cbnt.austin.ibm.com* is to be bound into the remote name context on the remote host.

**description**
An optional 256-character text field for you to store any text

For a more complete example, see Example Use of Remote Name Context Attributes (page 174).

d. To apply the changes and close the Object Editor window, click the **OK** button.

4. Configure the Remote Name Context onto the Hosts that are to perform the task of binding the remote name contexts. To do this, complete the following steps:

   a. On the pop-up menu of the Remote Name Context model, click **Drag**

   b. Display the Host model for a host that is to bind the remote name contexts

   c. On the pop-up menu of the Host model, click **Configure Remote Name Context**

   Repeat these steps for each host that is to bind the remote name contexts.

Repeat this procedure for each remote name context binding that you need to use on hosts in your Configuration.

When you next activate your Configuration, the hosts will bind the remote name contexts into their host name trees as specified by the attributes of the Remote Name Context models.

**Example Use of a Remote Name Context:**

Consider the following values specified for the attributes of a Remote Name Context:

| | |
|---|---|
| **remote host modifer** | iiop://big390.pok.ibm.com:900 |
| **remote name context** | .:/workgroups/cbworkstations/hosts |
| **local name context** | host |
| **binding direction** | local into remote |
| **binding name** | cbnt.austin.ibm.com |

This would result in the host tree for cbnt.austin.ibm.com to be bound as a member host into the work group named cbworkstations on the CB/390 host big390.pok.ibm.com.

**Related Concepts**

“Remote Name Context Connections to other System Name Trees” on page 23
“The System Name Tree” on page 21

## Configure a TCP/IP or IPC Protocol onto a Host

Use this procedure to configure a TCP/IP Protocol or IPC Protocol onto one or more Hosts. This specifies the hosts on which the protocol is to be used.

When you “Configure a new Host into your Network” on page 117, you configure a TCP/IP Protocol onto the new Host.

You only need to use this procedure if you want to configure another protocol onto existing hosts.

**Prerequisites:**

This task assumes that you have already defined the Protocol, as described in “Define a TCP/IP or IPC Protocol” on page 170.

**To configure a protocol onto a host, complete the following steps:**

1. Expand the Configuration of the Management Zone that you are using to configure your host network.
2. Expand the protocol’s folder; for example, TCP/IP Protocols.
3. On the protocol’s pop-up menu, click **Drag**.
4. Expand the Hosts Folder
5. On the pop-up menu of the Host, click **Configure ... Protocol**.

   This creates a *Configured ... Protocol* relationship (for example, Configured TCP/IP Protocol) between the Protocol and the Host that it has been configured on.

   Repeat this step for each Host on which the protocol is to be used.
6. Optionally, to verify that the protocol has been configured properly onto a host, complete the following steps:
   a. Expand the Host.
   b. Expand either the Configured TCP/IP Protocols folder or the Configured IPC Protocol folders, as appropriate.

   You should see a protocol object with the name of the configured Protocol (from 3). If not, repeat steps 3 to 5.

**Related Concepts**

“Hosts” on page 20

**Related Tasks**

“Define a TCP/IP or IPC Protocol” on page 170
“Configure a new Host into your Network” on page 117

## Change the DCE/CDS Confidence Level

By default, a name server reads data from the most convenient DCE/CDS server. However, you can configure the name server to get data from the CDS clerk’s cache or from the server where the master replica is located, depending on the degree to which the name server can trust the accuracy of data returned from the DCE/CDS server. You do this by changing the *DCE/CDS confidence level* of the name server.

To change the DCE/CDS confidence level setting for a name server, use the System Manager user interface to complete the following steps:

1. Expand the Management Zone that defines your host network
2. Expand the Configurations folder

3. Expand the Configuration within which the Name Server is defined
4. Expand the Name Servers folder
5. From the pop-up menu of the Name Server, click **Edit**. This displays the Object Editor window for the Name Server.
6. In the Object Editor Notebook, click the **Naming Service** tab.
7. Change the *DCE/CDS confidence level* to one of the following settings:

   **low** The name server reads from the CDS clerk's cache or from the most convenient DCE/CDS server.

   **medium**
   The name server reads from the most convenient DCE/CDS server.

   **high** The name server reads from the server where the master replica is located.

   **Default:** medium

   The low or medium confidence level is recommended for most applications. When a CDS server satisfies a request, the clerk refreshes its cache with the returned values and name server receives up-to-date data on future calls.
8. To save the changes and close the Object Editor Notebook, click the **OK** button.
9. To apply the changes to the name server in your enterprise, activate the Configuration.

   On the pop-up menu of the Configuration, click **Activate**.

   The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. You should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, you should see a *Activation successful* message.

## Remove a Name Server from a Host

Use this procedure to remove a name server from a host managed by the System Manager.

If you want to remove an individual application server from its host, complete the steps described in “Remove an Application Server from a Host” on page 196.

If you want to remove a complete controlled server group, complete the steps described in “Remove a Controlled Server Group from Your Enterprise” on page 341.

**Prerequisites:**

Before deleting a Name Server, any references to the Name Server's host must be removed from all Configurations in all Management Zones defined in the System Manager. The following relationships with the Name Server's host are examples of things to look for in the Configurations within each Management Zone:

- *Member Host*, *Preferring Host* and *Serving Host* relationships in Cells and Work Groups
- *Configured Host* relationships in Client Styles and TCP/IP protocols

After you have deleted these relationships from all Configurations, each Configuration needs to be activated so that the changes are reflected in the Image world. When you believe you have removed all references to a host from a Management Zone, you can check the *Used Hosts* list in the Management Zone to ensure that the host is no longer listed.

If you are uninstalling several hosts, the Name servers must be removed in the following order:

1. All iHostServices Name Servers
2. All iWorkGroupServices Name Servers
3. The iCellServices Name Server

The appropriate steps to take for deletion of a Name Server varies for iCellServices, iWorkGroupServices and iHostServices configured Name Servers. The specific steps to complete for each type follow:

**To remove an iHostServices Name Server, complete the following steps:**

1. Remove all references to the Name Server's host from the configuration, which at a minimum can be as follows:
2. 
   - *Member Host* and *Preferring Host* relationships for work groups
   - *Member Host* and *Preferring Host* relationships for a cell
   - *Configured Host* relationships for a TCP/IP Protocol
3. Activate the configuration
4. Verify that the host is not in the *Used Hosts* list for the Management Zone
5. From the pop-up menu of the Name Server Image, click **Remove**. The action console will inform you when the removal has been completed. If the action console indicates the server cannot be removed, you should force remove the server, as described in Force a Name Server to be removed (page 178).
6. To verify successful completion, use the DCE Director to check that the DCE CDS name space no longer contains any entries related to this host. If needed, use the DCE Director to remove any entries related to the host.

**To remove an iWorkGroupServices Name Server, complete the following steps:**

1. Determine which work groups have this host as the Serving Host.
2. For each identified work group, delete all *Member Host* relationships
3. Activate the configuration
4. Verify successful completion by using the DCE Director to check that the hosts context of each work group name tree has no links to host name trees
5. Delete the *Serving Host* relationships for the Name Server's host in all work groups except the work group that also has a *Preferring Host* relationship with that host
6. Activate the configuration
7. Verify successful completion by using the DCE Director to check the DCE CDS name space no longer contains any entries related to the work groups whose relationships were deleted in step 5. If needed, use the DCE Director to remove any entries related to the work groups.
8. Continue by completing the steps for an iHostServices name server (page 177)

**To remove an iCellServices Name Server, complete the following steps:**

**Note: This must be the last host in your Component Broker network from which a Name Server is removed**.

1. Delete from the Cell the *Member Host* relationship to the Name Server's host
2. Continue by completing the steps for an iWorkGroupServices name server

**Force a Name Server to be removed**

If the **Remove** on a Server Image fails to remove the name server, the action console indicates that the server is not being removed properly. You can remove the server forcefully as described in Remove Forcefully a Server from a Host. For example, if a server is recycling infinitely without progressing towards completing the server removal, consider removing the server forcefully.

**Related Tasks**

"Change the Active Configuration of Your Enterprise" on page 431
"Remove an Application Server from a Host" on page 196
"Remove a Controlled Server Group from Your Enterprise" on page 341
"Uninstall Managed Hosts"
"Activate a Configuration" on page 256
"Delete Objects from an Active Configuration" on page 259
"Control Which Objects are Displayed" on page 62

## Uninstall Managed Hosts

Use this procedure to uninstall Component Broker from one or more hosts managed by the System Manager.

**Before starting this procedure** read the follow notes:

- The general sequence of this procedure is to remove all servers from a host then run the Component Broker uninstall option on that host.
- You should remove servers from a host in the following order:
  1. All application servers
  2. Any SGCP servers and SGGW servers
  3. The name server on the host

  When uninstalling more than one host, the order in which the hosts are uninstalled is important. You should do the first step (remove application servers) for all affected hosts, then do the second step (remove SGCP/SGGW servers) for all affected hosts, then lastly remove the name servers for all the hosts.

  The name servers must be deleted in the following order:
  1. All iHostServices Name Servers
  2. All iWorkGroupServices Name Servers
  3. The iCellServices Name Server

  **Note: If you are uninstalling several hosts, the last host that you uninstall should be the host on which the cell name server runs.**

**To uninstall Component Broker from a host**, complete the following steps:

1. Remove all application servers from the host. If you want to remove a complete controlled server group, see "Remove a Controlled Server Group from Your

Enterprise” on page 341. Otherwise, to remove one or more application servers, see “Remove an Application Server from a Host” on page 196.

2. If they exist, remove any SGCP servers and SGGW servers from the host. This is described as part of the task “Remove a Controlled Server Group from Your Enterprise” on page 341.
3. Remove the name server from the host. See “Remove a Name Server from a Host” on page 176.
4. Uninstall Component Broker from the host; for example, by clicking **Start - Programs - Component Broker for Windows NT - Uninstall**.

### Related Tasks

“Change the Active Configuration of Your Enterprise” on page 431
“Remove an Application Server from a Host” on page 196
“Remove a Controlled Server Group from Your Enterprise” on page 341
“Remove a Name Server from a Host” on page 176

# Chapter 7. Administer Application Servers

To create application servers within your enterprise, you have to define and configure them within a system management Configuration. When you next activate the Configuration, the System Manager starts the servers running on their hosts.

You can define a server as either a separate, freestanding server or as a member of a server group.

- Use a freestanding server, defined by a *Server (free standing)*, if you want one server with unique characteristics. However, you can can convert a freestanding server into a server group at any time.
- Use a server group, defined by a *Server Group* and one or more *Servers (member of group)*, if you want several servers with the same characteristics. Typically, the identical server members of a server group support the same applications or implement workload management across the controlled server group.
  - The Server Group defines the attributes for all the servers that are members of its group.
  - Each Server (member of group) defines an application server that is a member of the group.

When you activate a Configuration of one of your application Management Zones, the System Manager updates the runtime configuration of the application servers defined in that Configuration and starts them on their hosts.

The servers are represented by *Server Images*, which can be accessed through the *Active Configuration* within the Management Zone and through the Host Images for hosts that the servers run on.

To change the runtime characteristics of individual application servers, you can act on their Server Images directly. However, the characteristics of a Server Image are reset when the system management Configuration containing the related Server model is next activated. To change the characteristics of application servers, you should change the Server Group or Server (freestanding) in a Configuration of a Management Zone, then activate that Configuration again.

For more information about configuring and operating application servers, see the following topics:

- “Define and Configure Servers and Server Groups” on page 182
- “Operate Application Servers” on page 192

**Related Concepts**

“The Model World” on page 39
“Application Servers and Server Groups” on page 25
“Controlled Server Groups” on page 336

**Related Tasks**

“Configure a new Server Group” on page 145
“Configure Applications onto a Server Group” on page 183
“Configure a new Freestanding Server” on page 149
“Configure an Application onto a Freestanding Server” on page 186
“Make AIX Shared Libraries Available to Application Servers” on page 191

"Chapter 9. Administer Applications" on page 221
"Chapter 11. Administer Security in your Enterprise" on page 261
"Chapter 12. Administer Workload Management" on page 333
"Chapter 13. Administer Connections to Tier-3 Systems" on page 343
"Chapter 14. Administer Component Broker Services" on page 397

## Define and Configure Servers and Server Groups

To create servers within your enterprise, you have to define and configure them within a system management Configuration. When you next activate the Configuration, the System Manager starts the servers running on their hosts.

You can define a server as either a separate, freestanding server or as a member of a server group.

- Use a freestanding server, defined by a *Server (free standing)*, if you want one server with unique characteristics. However, you can can convert a freestanding server into a server group at any time.
- Use a server group, defined by a *Server Group* and one or more *Servers (member of group)*, if you want several servers with the same characteristics. Typically, the identical server members of a server group support the same applications or implement workload management across the controlled server group.
  - The Server Group defines the attributes for all the servers that are members of its group.
  - Each Server (member of group) defines an application server that is a member of the group.

You can customize the characteristics of all servers in a server group by editing the attributes of the Server Group. You can customize the characteristics of a freestanding server by editing the attributes of the Server (freestanding). There is a large set of attributes that you can customize to match your needs. The attributes that you might normally want to change are described in other task topics. Information about each attribute is provided for context-sensitive help through the System Manager user interface.

Before you first activate a Configuration, you should decide whether each Server Group that it contains is to be always a controlled server group (for workload management) or basic server group, and configure the Server Group accordingly. (You should not later switch any server group from its original controlled or basic configuration.)

This topic describes the following tasks:

- "Configure a new Server Group" on page 145
- "Configure Applications onto a Server Group" on page 183
- "Configure a new Server as a Member of a Server Group" on page 185
- "Configure a new Freestanding Server" on page 149
- "Configure an Application onto a Freestanding Server" on page 186
- "Configure the Attributes of a Server or Server Group" on page 187
- "Configure how Application Servers are to Run" on page 187
- "Configure Security for a Server" on page 323

- “Configure a Server to use the Query Service” on page 402
- “Configure a Server Group to use the Transaction Service” on page 403
- “Configure a Freestanding Application Server to use the Transaction Service” on page 405
- “Configure a Server to Provide Extended Component Broker Services” on page 406
- Move an Application Server to run on a Different Host
- “Make a Server Group from a Free Standing Server” on page 189
- “Make a Free Standing Server From a Server Group” on page 190

**Related Concepts**

“The Model World” on page 39
“Application Servers and Server Groups” on page 25
“Controlled Server Groups” on page 336
Planning Your System Management Network
“Wizards” on page 447

**Related Tasks**

“Example: Change the Values of Attributes” on page 198
“Chapter 9. Administer Applications” on page 221
“Chapter 11. Administer Security in your Enterprise” on page 261
“Chapter 12. Administer Workload Management” on page 333
“Chapter 13. Administer Connections to Tier-3 Systems” on page 343
“Chapter 14. Administer Component Broker Services” on page 397

## Configure Applications onto a Server Group

Use this task to configure applications onto a server group within an existing Configuration of one of your application Management Zones.

To configure one or more applications onto a server group in your application environment, you can use the **Configure Servers** wizard, as described in this topic.

**To configure one or more applications onto a server group, complete the following steps:**

1. On the menu-bar of the System Manager user interface, click **Wizards - Configure Servers**.
2. On the Select Applications to Configure panel, select the applications that you want to configure onto the server group.

   To add an application to the list to be configured, displayed under *Applications to Configure*, complete the following steps:

   a. In the *Available Applications* list box, click on one or more applications. To select several applications, press and hold the Ctrl key while clicking on the applications then release the Ctrl key.
   b. Click the **Add** button.

   The applications are added to the list of applications to be configured on the server group, listed in the *Applications to Configure* list box.

To remove an application from the *Applications to Configure* list box, complete the following steps:

a. Select the application's entry in the *Applications to Configure* list box.
b. Click the **Remove** button.

The application is removed from the list of applications to be configured on the server group.

3. On the Management Zone panel, select the application Management Zone within which the server group is defined.
4. On the Configuration panel, select the Configuration (of the Management Zone that you selected previously) within which the server group is defined.
5. On the Server Group panel, select the name of one or more server groups onto which the applications are to be configured.

   To add a server group to the list under *Servers To Configure Applications On*, complete the following steps:

   a. In the *Available Server Groups* list box, click on one or more server groups. To select several server groups, press and hold the Ctrl key while clicking on the server groups then release the Ctrl key.
   b. Click the **Add** button.

   The server groups are added to the list of server groups in the *Servers To Configure Applications On* list box.

   To remove a server group from the *Servers To Configure Applications On* list box, complete the following steps:

   a. Select the server group's entry in the *Servers To Configure Applications On* list box.
   b. Click the **Remove** button.

   The server group is removed from the list of server groups onto which the selected applications are to be configured.
6. To finish and have the wizard configure the applications onto the server group, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

   If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.
7. To leave the wizard without having it perform any action, click the **cancel** button on any panel.

**Note:** The wizard does not change the server group until you click the **finish** button on any panel and have specified the details that it needs. This updates the server group within the Configuration specified, but does not change the servers that are members of the server group in your enterprise.

The servers in your enterprise are changed the next time that you activate the Configuration.

**Related Concepts**

"Workload Management" on page 335

“Application Servers and Server Groups” on page 25
“Controlled Server Groups” on page 336
“Wizards” on page 447

**Related Tasks**

“Configure a new Server Group” on page 145
“Chapter 5. Configure a new Application Environment” on page 133

## Configure a new Server as a Member of a Server Group

To configure an application server as a member of a server group, you can use the **Create Servers** wizard, as described in “Configure a new Server Group” on page 145.

Alternatively, you can configure a Server (member of group) manually, within a Server Group that has already been created.

Both methods create a Server (member of group) to define the server and configure the Server (member of group) onto the Host on which the application server is to run.

Applications to run on the server are configured onto the Server Group, as described in “Configure Applications onto a Server Group” on page 183.

**To configure a Server (member of group) manually, complete the following steps:**

1. Display the Server Group that the Server (member of group) is to be a member of.
2. On the pop-up menu of the Server Group, click **New - Member of group**.

   This displays a dialog box for you to specify a unique name for the new application server.
3. Type a unique name for the Server (member of group). The name can contain:
   - From 1 through 32 ASCII characters; a through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks

   **Note:** A Server Model cannot have the same name as any other Server Model or any Server Group model in the same Configuration. (See Names of Server Groups and Servers.)
4. To create the object, click the **OK** button. If the name is valid, this creates an object in the Servers (member of group) folder within the Server Group. If the name is not valid, you are prompted to enter a new name.

   To cancel the procedure, click the **Cancel** button.
5. Expand the Servers (member of group) folder. You should see an object with the name specified in step 3. If not, repeat steps 2 to 4.
6. On the pop-up menu of the Server (member of group), click **Drag**.
7. Expand the Hosts folder
8. On the pop-up menu of the Host that the server is to run on, click **Configure server**. This creates a relationship between the Server (member of group) and the Host that it has been configured on.

9. Optionally, to verify that the relationship has been created, complete the following steps:
   a. Expand the Server (member of group).
   b. Expand the *Configured Hosts* folder You should see a Host object with the name of the Host configured. If not, repeat steps 6 to 8.

**Related Concepts**

“Application Servers and Server Groups” on page 25

**Related Tasks**

“Example: Configure a new Multi-host Application Environment” on page 159
“Configure the Attributes of a Server or Server Group” on page 187
“Configure Applications onto a Server Group” on page 183
“Define and Configure Servers and Server Groups” on page 182
“Operate Application Servers” on page 192
“Chapter 5. Configure a new Application Environment” on page 133

## Configure an Application onto a Freestanding Server

Use this task to configure applications onto a freestanding server within an existing Configuration of one of your application Management Zones.

If you want to configure applications onto a server group, see “Configure Applications onto a Server Group” on page 183.

**Prerequisites:** This task assumes that the application and freestanding server have both been defined in the same Configuration of your application Management Zone. If not, see one or both of the following topics:

- If you have not already defined the application, see “Load a new Application” on page 141.
- If you have not already defined the freestanding server, see “Configure a new Freestanding Server” on page 149, which includes steps to configure applications onto the new server.

**To configure one or more applications onto a freestanding server, complete the following steps:**

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Expand the Management Zones folder
3. Expand your application Management Zone.
4. Expand the Configurations folder of your Management Zone.
5. Expand the Configuration that contains the Server (freestanding).
6. Expand the Applications folder. You should see the Application that you want to configure on a server.
7. On the Application’s pop-up menu, click **Drag**.
8. Expand the Server (free standing) folder.
9. On the pop-up menu of theServer (free standing), click **Configure Application**. This creates a *Configured Applications* relationship between the Application and the Server (free standing) that it has been configured on.

10. Optionally, to verify that the relationship has been created, expand the Server (free standing). You should see a *Configured Applications* relationship with the name of the configured Application (from 6). If not, repeat steps 7 to 9.

**Note:** You cannot configure applications from the Available Applications folder directly onto Server Groups or Servers (free standing). You first “Add an Application into a Configuration of your Application Environment” on page 228 then configure the Application onto the Server Groups or Servers (freestanding) as required.

**Related Concepts**

“Applications” on page 27
“The Model World” on page 39
Planning Your System Management Network

**Related Tasks**

“Add an Application into a Configuration of your Application Environment” on page 228

## Configure the Attributes of a Server or Server Group

When you create a new Server Group, Server (member of group), or Server (freestanding server), it is configured automatically with default attribute values. The default values may not be suitable for your needs.

You can change the attributes of a Server Group or Server (freestanding) by completing tasks such as those listed in the related tasks at the end of this topic or for information provided with applications for Component Broker. In the descriptions of such tasks you are instructed to change attribute values as required.

**Note:** You generally change the attributes of a Server (member of group) on its Server Group. The changes apply to all Servers (member of group) within the Server Group.

If you otherwise need to change the attributes, you **Edit** the Server Group or Server (freestanding). This displays an Object Editor window for the Server or Server Group. (See “Edit Objects” on page 72.)

**Related Tasks**

“Configure how Application Servers are to Run”
“Configure a Controlled Server Group” on page 339
“Configure a Server to use the Query Service” on page 402
“Configure a Server Group to use the Transaction Service” on page 403
“Configure a Freestanding Application Server to use the Transaction Service” on page 405
“Configure a Server to Provide Extended Component Broker Services” on page 406
“Configure Security for a Server” on page 323
“Configure a Server for Connections to tier-3 Systems” on page 392
“Change the userid and password used to access a DB2 Database” on page 243

## Configure how Application Servers are to Run

Use this procedure to configure how application servers are to run.

Before you activate a system management Configuration, you would normally configure how application servers are to run. By default, an application server is configured automatically to run immediately when its system management Configuration is activated.

You can configure how an application server is to run by setting its **run control** attribute to `run on request` or `run immediate` (the default), as described in this procedure.

The value of the **run control** attribute on a Server Group sets the intended method for running all application servers in the server group. The value for a Server (freestanding) sets the intended method for running that application server only.

After the system management Configuration has been activated, you can change the run control status ofan application server, by using the **run** or **run immediate** action on its Server Image. However, for long-term changes, you should change the **run control** attribute of the Server Group or Server (freestanding) as described in the procedure.

**To configure how an application server is to run, complete the following steps:**

1. Start the System Manager User Interface, and return to the home view; for example, by clicking the 🏠 icon of the Tool bar.
2. Expand the Management Zones folder
3. Expand the Management Zone within which the application server is defined
4. Expand the Configuration within which the application server is defined
5. Display the Object Editor.

   For a server group, complete the following steps:

   a. Expand the Server Groups folder

   b. From the pop-up menu of the Server Group that the application server is a member of, click **Edit**.

   For a freestanding server

   a. Expand the Servers (freestanding) folder

   b. From the pop-up menu of the Server (freestanding) that defines the application server, click **Edit**.
6. Select the **Main** notebook page tab.
7. Select one of the following options for the **run control** attribute, as required:

**run on request**

The server is started by the first ORB request that it receives. This minimizes the host resources used for the server until it receives an ORB request, but incurs a delay for such requests.

**run immediate**

The server is started immediately its system management Configuration is activated.

**stop**

The server is stopped and cannot be started until its **run control** status is later set to `run on request` or `run immediate`.

The other options, `stop quick` and `stop immediate`, are intended to represent actions taken to stop an application server having difficulties. They are normally set automatically by actions on a Server Image, and should not be used when configuring how application servers are to run.

8. To apply the change, and close the Object Editor window, click **OK**.
9. To check that Configuration is valid and update the application servers with their methods of running, activate the Configuration.

   On the pop-up menu of the Configuration, click **Activate**.

   The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

**Note:** If you set an application server's **run control** attribute to `run on request`, then you should increase the value of the **request timeout** attribute for all clients of that server. The default request time out value for client styles is too short when an application server's **run control** attribute is set to `run on request`. This is because a run on request application server must be started when it receives its first request. The response time for this first request is therefore much longer than the response time for subsequent requests.

**Related Concepts**

"Application Servers and Server Groups" on page 25

**Related Tasks**

"Configure the Request Timeout for Client Styles" on page 214
"Start a Server" on page 192
"Configure a new Server Group" on page 145
"Configure a new Server as a Member of a Server Group" on page 185
"Configure a new Freestanding Server" on page 149
"Chapter 7. Administer Application Servers" on page 181

## Make a Server Group from a Free Standing Server

Use this procedure to create a Server Group from a Server (free standing). Besides creating the new Server Group, this procedure creates a Server (member of group) for the Server (free standing) and configures the applications onto the Server Group.

1. On the pop-up menu of the Server (free standing), click **Convert to Group**. This displays a dialog box for you to specify a unique name for the Server Group.
2. Type the name that you want the new Server Group to be known by. The name can contain:
   - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks

   (See Names of Server Groups and Servers.)
3. To create the Server (free standing), click the **OK** button. If the name is valid, this creates the Server (free standing). If the name is not valid, a dialog box is displayed for you to enter a new name.

   To cancel the procedure, click the **Cancel** button.

4. Optionally, to verify that the Server Group has been created, expand the Server Groups folder.

   You should see an object with the name specified in step 2. If not, repeat steps 1 to 3.

**Related Concepts**

"Application Servers and Server Groups" on page 25
"Controlled Server Groups" on page 336

**Related Tasks**

"Example: Configure a new Multi-host Application Environment" on page 159
"Configure a new Server Group" on page 145
"Configure Applications onto a Server Group" on page 183
"Define and Configure Servers and Server Groups" on page 182
"Operate Application Servers" on page 192
"Make a Free Standing Server From a Server Group"

## Make a Free Standing Server From a Server Group

Use this procedure to create a Server (free standing) model from a Server (member of group) model.

1. On the pop-up menu of the Server (member of group) model, click **Convert to Free Standing**. This displays a dialog box for you to specify a unique name for the Server (free standing).
2. Type the name that you want the new Server (free standing) to be known by. The name can contain:
   - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks

   (See Names of Server Groups and Servers.)
3. To create the Server (free standing) model, click the **OK** button. If the name is valid, this creates the Server (free standing) model. If the name is not valid, a dialog box is displayed for you to enter a new name.

   To cancel the procedure, click the **Cancel** button.
4. Optionally, to verify that the Server (free standing) model has been created, expand the Servers (free standing) folder.

   You should see an object with the name specified in step 2. If not, repeat steps 1 to 3.

Besides creating a Server (free standing) model, the Application models configured on the Server Group model are reconfigured automatically onto the Server (free standing) model.

**Related Concepts**

"Application Servers and Server Groups" on page 25

**Related Tasks**

"Example: Configure a new Multi-host Application Environment" on page 159
"Configure a new Server Group" on page 145
"Configure an Application onto a Freestanding Server" on page 186
"Define and Configure Servers and Server Groups" on page 182

"Operate Application Servers" on page 192
"Make a Free Standing Server From a Server Group" on page 190

## Make AIX Shared Libraries Available to Application Servers

AIX Use this procedure to enable application servers on an AIX server host to access shared libraries that are not in one of the directories specified in the LIBPATH environment variable for that host.

**Considerations:**

- You may need to use this procedure for shared libraries that are not loaded as part of a Component Broker application family package. When applications are loaded by their installation tool or the **Load Application** action of the System Manager user interface, Component Broker automatically updates the LIBPATH environment variable before starting application servers.
- When you have completed this procedure, the shared libraries will be available to all application servers on the same host, even if you later stop and restart them.

If an application server needs to use shared libraries that are not in one of the directories specified in the LIBPATH, you must make those libraries available before the application server is started. To do this, you can take either of the following actions:

- Ensure that the shared libraries for the application servers are in one of the directories specified in the CBConnector.profile.
- Modify the LIBPATH environment variable to include the directories that contain the shared libraries

If you modify the CBConnector.profile, you must stop and restart the CBConnector system management service (bgmain) on the server host to pick up the changes. You can do this by completing the following steps:

1. Login as root on the AIX server host.
2. **To stop the CBConnector system management process, complete the following steps:**
   a. From a shell prompt, enter the following:
      ```
      smitty apps
      ```
   b. Move the cursor to **Stop the System Management Application** then press Enter.
   c. On the Stop the System Management Application now? window, press the Tab key to change the response to `y` then press Enter.
3. **To restart the CBConnector system management service, complete the following steps:**
   a. From a shell prompt, enter the following:
      ```
      smitty apps
      ```
   b. Move the cursor to **Start the System Management Application** then press Enter.

**Related Concepts**

"Installing Applications" on page 222
"Applications" on page 27

**Related Tasks**

"Chapter 7. Administer Application Servers" on page 181
"Chapter 9. Administer Applications" on page 221

## Operate Application Servers

When you activate a Configuration of one of your application Management Zones, the System Manager updates the runtime configuration of the application servers defined in that Configuration and starts them on their hosts.

The servers are represented by *Server Images*, which can be accessed through the *Active Configuration* within the Management Zone and within the Host Images for hosts that the servers run on.

This topic describes how to do some general operation tasks for servers, which affect the current runtime configuration of the servers. For example, it describes tasks to start and stop servers running on managed hosts.

To change the runtime characteristics of individual application servers, you can act on their Server Images directly. However, the characteristics of a Server Image is reset when the system management Configuration containing the related Server model is next activated. To change the active configuration of application servers, you change the Server Group or Servers (freestanding) in a Configuration of a Management Zone, then activate that Configuration again, as described in "Define and Configure Servers and Server Groups" on page 182.

This topic describes:

- "Start a Server"
- "Stop a Server" on page 193
- "Display Server Health" on page 195
- "Display Server Container Statistics" on page 195
- "Display Event Channel Statistics" on page 196
- "Remove an Application Server from a Host" on page 196

For information about other operation concepts and tasks, see "Chapter 15. Operate your Enterprise" on page 413.

**Related Concepts**

"Example: Operate a Server" on page 201

### Start a Server

When you activate a Configuration, all the servers defined in that configuration are started automatically, so that the Name Tree is updated correctly for the server and its contents. After their Configuration has been activated, individual servers can be stopped and restarted as needed. Normally, the servers would be restarted by activating the Configuration again. Otherwise, the procedure described in this topic can be used.

You start a transactional server just as you would start any other server. When it starts, it produces a number of messages, as described in Types of Server Start-up.

Use this procedure to restart a server running *after it has been started at least once by activating its Configuration*.

Note: AIX If an application server needs to use shared libraries that are not in one of the directories specified in the LIBPATH, you must make those libraries available before the application server is started. For information about how to do this, see “Make AIX Shared Libraries Available to Application Servers” on page 191.

You can restart a server so that it runs immediately or so that it waits until it receives an ORB request to run.

- To restart a server so that it waits until it receives an ORB request to run, complete the following steps:
  1. From the pop-up menu of the Server Image, click **stop** (if it is not stopped already)
  2. Ensure that the **request timeout** value for all clients of the server is big enough. (See Note 2 below.)
  3. From the pop-up menu of the Server Image, click **run**

  This minimizes the host resources used for the server until it receives an ORB request, but incurs a delay for such requests.
- To restart a server so that it runs immediately, complete the following steps:
  1. From the pop-up menu of the Server Image, click **stop** (if it is not stopped already)
  2. From the pop-up menu of the Server Image, click **run immediate**

  This enables the server to process its first ORB method request without delay.

**Notes:**

1. If the **stop** action fails to stop a server, see “Stop a Server”.
2. The default request time out value for client styles is too short when the application server’s **run control** attribute is set to `run on request`. This is because a run on request application server must be started when it receives its first request. The response time for this first request is therefore much longer than the response time for subsequent requests.

   Therefore, before you use the **run** action on a Server Image, you should ensure that the value of the **request timeout** attribute for all clients of that server is sufficiently big.

**Related Tasks**

“Configure the Request Timeout for Client Styles” on page 214
“Activate a Configuration” on page 256
“Stop a Server”
“Display Server Health” on page 195

## Stop a Server

Use this procedure to stop a server running. This procedure can use any of the following alternative actions:

**Stop** Use this action to stop the server normally. This action:

- Waits for user tasks to complete on the server
- Performs any shutdown processing for the server
- Writes information to the CBConnector System Management logs

**Stop Quick**
Use this action to stop the server quickly with minimal shutdown processing. This action should be used only if the **Stop** action has failed to stop the server, because this action *does not* wait for user tasks to complete on the server.

**Stop Immediate**
Use this action to stop the server immediately with no shutdown processing. Use this action only as a method of last resort, if a **Stop** action or **Stop Quick** action has failed to stop the server.

To stop a server, complete the following steps:

1. Display the Server Image
2. On the pop-up menu of the Server Image, click **Stop**, **Stop Quick**, or **Stop Immediate**, as required.

When the stop action begins, an action console window is displayed to show messages issued as the action progresses. These messages are also logged in the activity log for the server's host.

If you need to shut down a server in a forced manner using the **Stop Quick** or **Stop Immediate** action, connections and locks can be left on your DB2 database and can interfere with future runs of the server. Therefore, you may need to manually clean up DB2 connections and locks that the server obtained while running.

To manually clean up after an untimely or forced shutdown, the following tasks may need to be done from a DB2 command prompt. To open a DB2 command window from the Windows NT **Start** menu, click **Programs - DB2 for Windows NT - DB2 Command Line Processor**.

- DB2 connections that continue to exist after shut down can cause a problem because there is a limit to how many connections DB2 allows concurrently. Connections can be listed by entering the command **list applications** from a DB2 command prompt. After a shut down, there should be no connections for the server that has been shutdown. If any connections are listed for the server, use the DB2 **force applications** command to remove them.
- Locks on DB2 data held by transactions that were ″in flight″ when the server was shut down can cause a problem because they can lock out other accesses to the data. To free these locks, use the DB2 **force applications** command.
- Locks on DB2 data held by transactions that were ″in doubt″ when the server was shut down can cause a problem because they can lock out other accesses to the data. Locks that are ″in doubt″ can be listed by entering **list indoubt transactions when prompting** from a DB2 command prompt. If any transactions are listed numerically, commit or rollback the transactions with commands such as **c 1** or **r 1**

A full discussion of these commands can be found in the *DB2 Command Reference* manual installed with DB2. For an in-depth discussion of manual recovery, see the *DB2 Administration Guide*.

**Related Concepts**

"Application Servers and Server Groups" on page 25

**Related Tasks**

"Activate a Configuration" on page 256
"Start a Server" on page 192
"Display Server Health"
"Operate Application Servers" on page 192
"Define and Configure Servers and Server Groups" on page 182

## Display Server Health

To display the health of an application server, select its Server Image. This causes the server health to be displayed on the Status bar of the Information Controller window. For example, for the server **My Server 1** running with good health, the Status bar would display:

My Server 1: runOnRequest running 3 3 **good**

Alternatively, to display the health of a server in an Object Editor window, complete the following steps:

1. On the pop-up menu of the Server Image, click **Edit**. This displays an Object Editor for the Server Image.
2. In the Object Editor window, click on the **Main** tab

This displays the **health** attribute, which has one of the following relative values:

| excellent | The best health value |
|---|---|
| **good** | Worse than **excellent**, but better than **OK** |
| **OK** | Worse than **OK**, but better than **poor** |
| **poor** | The worst health value, either because the server is heavily loaded or, before the server is started, because CBConnector System Management cannot determine the health value |

Default: poor

**Related Concepts**

"Monitoring the Health of Application Servers" on page 7
"Application Servers and Server Groups" on page 25
"The Image World" on page 40

**Related Tasks**

"Stop a Server" on page 193
"Activate a Configuration" on page 256

## Display Server Container Statistics

To display the statistics about the use of a server container, complete the following steps:

1. Display the Container Image.
2. On the pop-up menu of the Container Image, click **Edit**. This displays an Object Editor for the Container Image.

3. In the Object Editor window, click on the **Statistics** tab.

This displays statistics attributes for the container; for example, the **number of objects looked up** attribute.

For information about these attributes, see the context-sensitive help information provided by the System Manager user interface. To do this, click in an attribute field, then press the F1 key.

**Related Tasks**

“Display Event Channel Statistics”
“Operate Application Servers” on page 192

## Display Event Channel Statistics

To display the statistics about the use of an event channel, complete the following steps:

1. Display the Event Channel Image.
2. On the pop-up menu of the Event Channel Image, click **Edit**. This displays an Object Editor for the Event Channel Image.
3. In the Object Editor window, click on the **Main** tab.

This displays the statistics attributes for the event channel; for example, the **channel pulls per second** attribute.

For information about these attributes, see the context-sensitive help information provided by the System Manager user interface. To do this, click in an attribute field, then press the F1 key.

**Related Tasks**

“Display Server Container Statistics” on page 195
“Operate Application Servers” on page 192

## Remove an Application Server from a Host

Use this procedure to remove an application server from a host managed by the System Manager. This also removes the server’s entries from enterprise services such as DCE and the Host Naming Service.

**Note:** You can remove several servers at the same time, by selecting their system management objects then “Act on Several Selected Objects” on page 68. For example, to delete several Server models, switch to List view, select those models, then click **Selected - Delete** from the menu bar.

If you want to remove a complete controlled server group, complete the steps described in “Remove a Controlled Server Group from Your Enterprise” on page 341.

If you want to remove a name server from a host, complete the steps described in “Remove a Name Server from a Host” on page 176.

**To remove a server, complete the following steps:**

1. Delete the model that defines the server from its Configuration then reactivate the Configuration, as described in “Delete Objects from an Active Configuration” on page 259.

   This updates the active configuration and stops the server to be removed. An Action Console window is displayed where you can monitor the progress of the action. When a server is able to be removed, the console displays a message ″*xxx server has been deactivated and can now be deleted*″. Note that before this message is displayed, servers being deactivated may be started then stopped one or more times if needed.

   When the activate action has completed, as indicated by a completion message in the Action Console window, you can remove the server as follows.

2. All the configured applications in the server should have been removed from the server, which results in removal of registered objects (such as homes) from the name tree. To ensure that this process was successful, complete the follow checks:

3. 
   - In the Server Image, check the Application Images to see if any of your configured applications are still there (iObjectServices should be the only application after the process has completed).
   - Using the DCE Director, go to the local root for this host and check the path host/resources/servers/<servername> and see if there are any Homes bound into this context that are from your applications (note that there will be some still there from the iObjectServices application)
   - Using the DCE Director, go to the local root for this host and check the path host/resources/factories/somlcRepository/serverBranch/<servername>/collections and see if there are any Homes bound into this context that are from your applications (note that there will be some still there from the iObjectServices application)

   If the above checks show your applications have been deleted from the server, it is now safe to remove the Server Image, as follows:

4. Display the Server Image. (To do this, you may need to “Control Which Objects are Displayed” on page 62 to **Expert**.)

5. On the pop-up menu of the Server Image, click **Remove**.

   This has the following effects:

   - It starts an asynchronous action to remove the server from its host and displays an Action Console window to show the progress of this action.
   - It removes the Server Image from the Server Images folder and creates a corresponding object with the same name in the *Server Images (being removed)* folder to indicate that the server is in the process of deletion.

   Wait until the Remove action has completed before removing any other servers from that host. The action console will tell you when this process has completed.

   If the action console indicates that a server is not being removed properly, you can remove the server forcefully as described in Remove Forcefully a Server from a Host. For example, if a server is recycling infinitely without progressing towards the completion of removal of server, consider removing the server forcefully.

Besides checking the changes made by this task, you can use the DCE Director to check that the server name (<servername>) is not bound into the context host/resources/servers and the context host/resources/factories/somlcRepository/serverBranch.

**Related Tasks**

"Change the Active Configuration of Your Enterprise" on page 431
"Remove a Controlled Server Group from Your Enterprise" on page 341
"Remove a Name Server from a Host" on page 176
Remove Forcefully a Server from a Host
"Uninstall Managed Hosts" on page 178
"Activate a Configuration" on page 256
"Delete Objects from an Active Configuration" on page 259
"Control Which Objects are Displayed" on page 62
"Act on Several Selected Objects" on page 68

## Example: Change the Values of Attributes

This example guides you through changing the **Operating Mode** attribute of the Server models called *My Server 1* and *My Server 2*. It illustrates the general process for changing attributes, typically used after defining a server to tailor its characteristics.

It assumes that you have the Information Controller window displayed to show the Server Group called *My Server Group*, as shown in Figure 40 on page 199.

**Note:** Because the servers are defined as members of a server group, the attributes are set (and changed) on the Server Group model. The Server (member of group) models only identify the members of the server group and the hosts on which those servers are to run. This displays the Object Editor as shown in Figure 41 on page 199.

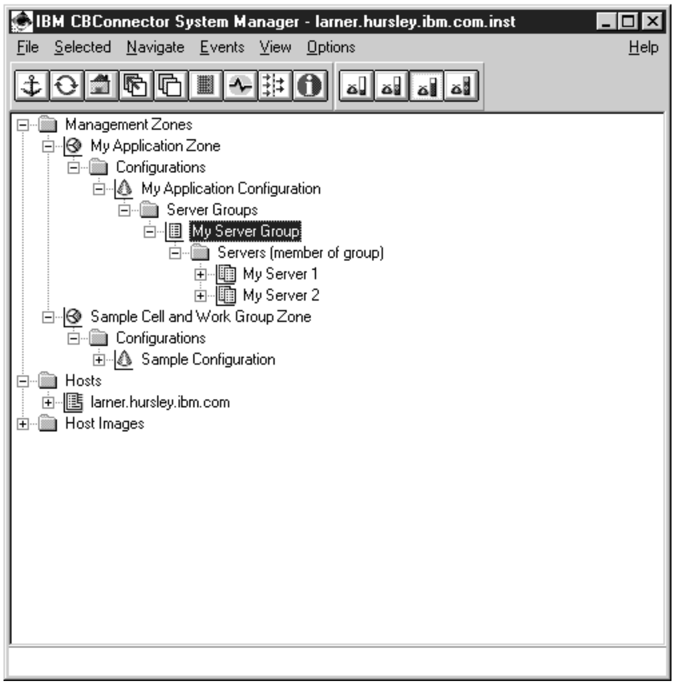
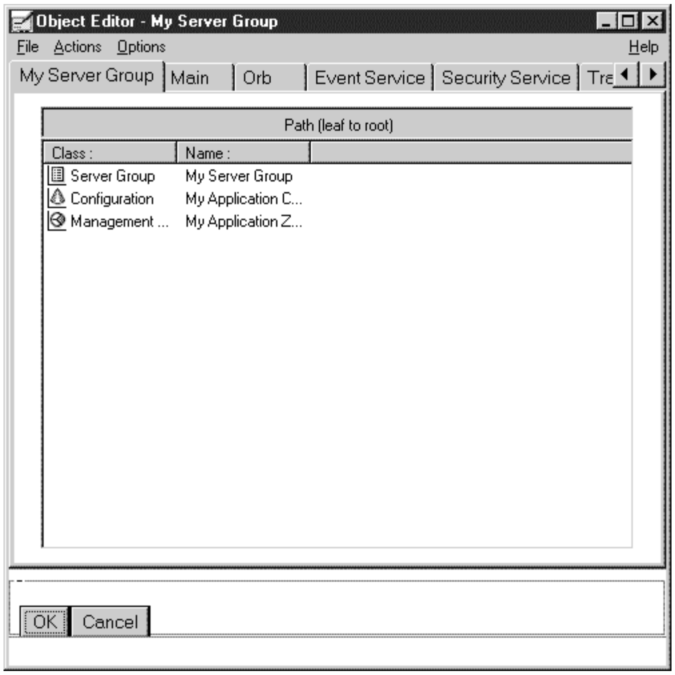

IBM CBConnector System Manager - larner.hursley.ibm.com.inst
File Selected Navigate Events View Options Help
Management Zones
My Application Zone
Configurations
My Application Configuration
Server Groups
My Server Group
Servers (member of group)
My Server 1
My Server 2
Sample Cell and Work Group Zone
Configurations
Sample Configuration
Hosts
larner.hursley.ibm.com
Host Images

*Figure 40. The Information Controller window.*
*Showing the example Server Group and its Servers (member of group)*

To change the value of the Server models' **run control** attribute, you use the Object Editor window as follows:

1. To display the Object Editor, on the pop-up menu of the Server Group **My Server Group**, click **edit .** This displays the Object Editor, as shown in the following figure:

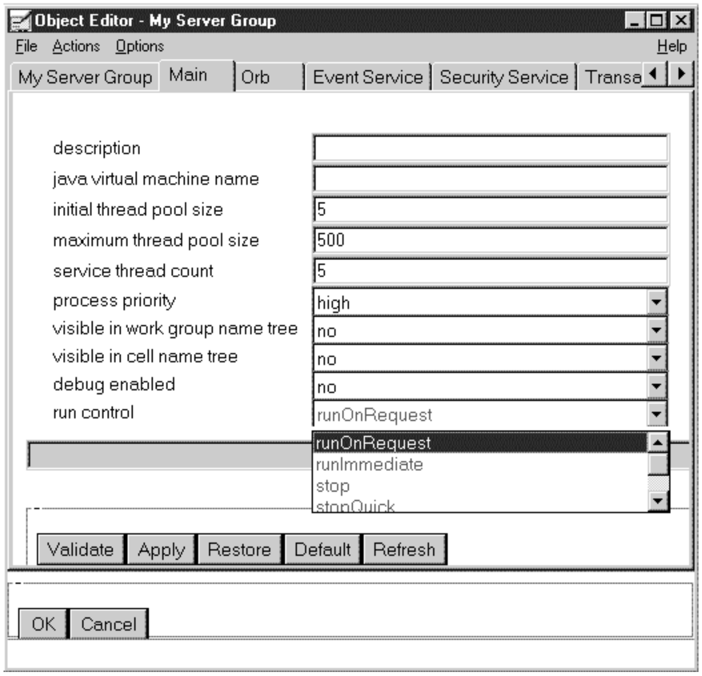


| Class : | Name : |
|---|---|
| Server Group | My Server Group |
| Configuration | My Application C... |
| Management ... | My Application Z... |

*Figure 41. The Object Editor window for a Server Group model.*

2. The Object Editor displays a notebook that shows a title page for the object, and some tabs along the top of the notebook for you to select alternative pages of attributes.
3. Click on the **Main** tab at the top of the notebook.
4. To display the pull-down options menu for the **run control** attribute, click on the pull-down mark on the right of its value field.

   This displays the pull-down options menu as shown in the following figure:



| | |
|---|---|
| description | |
| java virtual machine name | |
| initial thread pool size | 5 |
| maximum thread pool size | 500 |
| service thread count | 5 |
| process priority | high |
| visible in work group name tree | no |
| visible in cell name tree | no |
| debug enabled | no |
| run control | runOnRequest |

*Figure 42. The Object Editor window for a Server Group.*
*Showing the pull-down options menu for the run control attribute*

5. Click on the **runImmediate** value.

   **Note:** When you change any attribute, the attribute is marked by changing its color. This is to help you identify attributes that you have changed.
6. To apply the change and keep the Object Editor window displayed, click the **Apply** button. Note that the attribute is no longer marked, and has returned to its normal color. (Alternatively, to apply the change and close the Object Editor window with one action, you could have clicked the **OK** button.)
7. To return the attribute to its default value, click the **Default** button. Note that the value returns to **runOnRequest**.
8. To close the Object Editor (and commit any changes made), click the **OK** button.

Although you can change the attributes of a Server Group model at any time, it is usual to make most initial changes before configuring applications onto a server group.

### Related Concepts

An Example Tour Of CBConnector System Management
The Example System Management Network And Enterprise

**Related Tasks**

“Edit Objects” on page 72
“Example: Install an Application” on page 244
“Example: Define an Application” on page 246
“Example: Configure Applications on Servers” on page 248
“Example: Operate an Application” on page 250
“Example: Operate a Server”

## Example: Operate a Server

This example guides you through using the Information Controller window to operate the server, **My Server 2**, through its Server Image.

It assumes that you have the Information Controller window displayed to show the Server Image as shown in Figure 43.

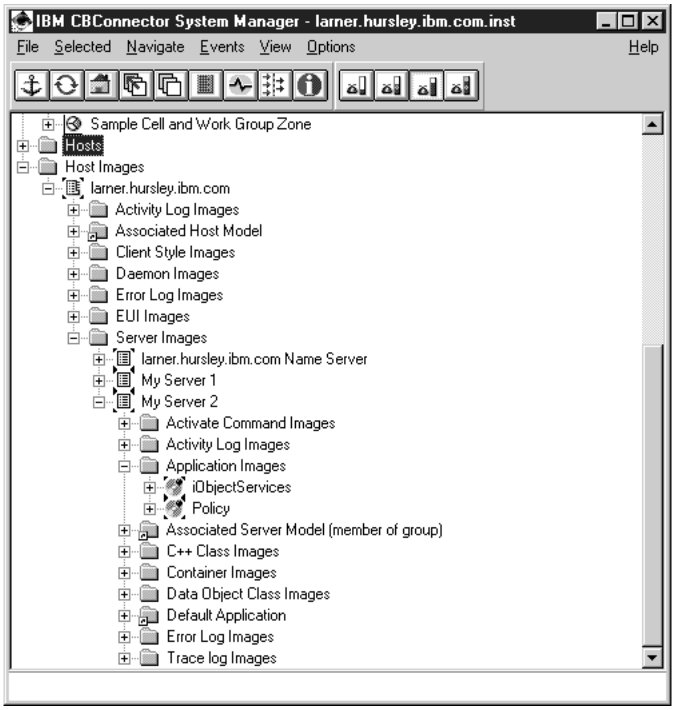


*Figure 43. The Information Controller window. Showing the Server Image, My Server 2*

This topic describes some examples task for operating a server. The tasks covered are to do the following:

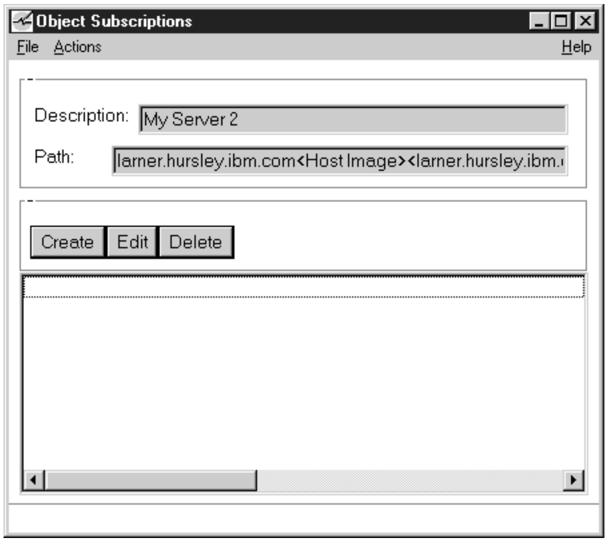
1. Subscribe to an event
2. Stop the server
3. Start the server

## Subscribing to an event

This task uses the Object Subscriptions window to subscribe to an event for the Server Image **My Server 2**.

1. Click the right mouse button on **My Server 2**, to display its pop-up menu

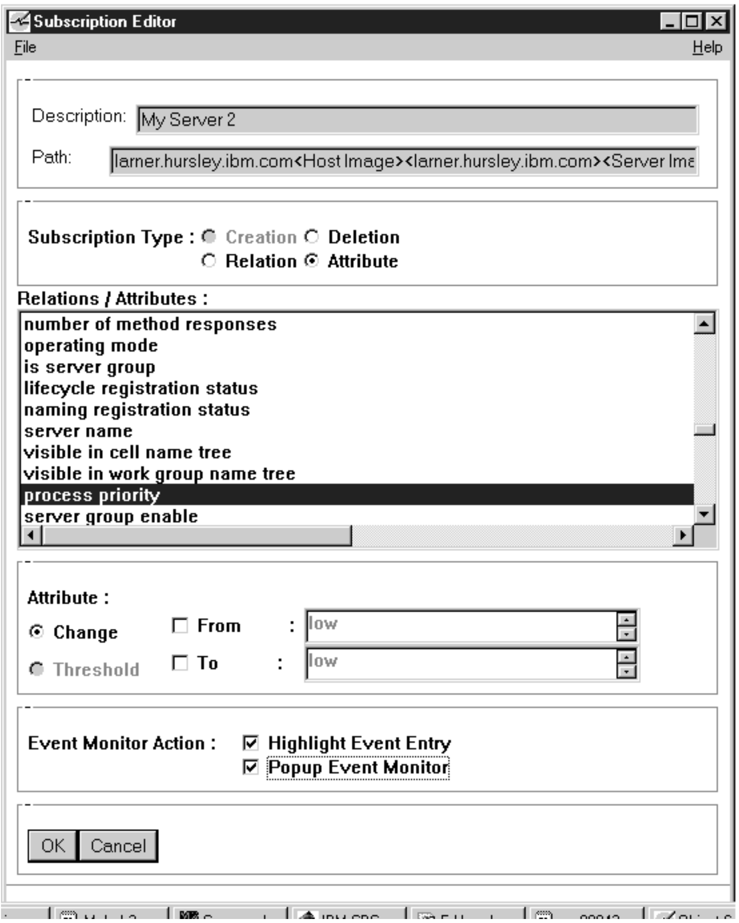
2. Click on **Subscriptions**, to display the Object Subscriptions window



Description: My Server 2

Path: larner.hursley.ibm.com<Host Image><larner.hursley.ibm.

3. In the Object Subscriptions window, click on the **Create** button, to display the Subscription Editor window
4. In the **Subscription Type** field, click on the **Attributes** radio button, to specify that you want to subscribe to an event to do with an attribute of **My Server 2**. This displays a list of attributes in the **Relationships/Attributes** field.
5. Scroll the Relationships/Attributes field down about half way to the bottom, to display the **process priority** attribute
6. Click **process priority**
7. You can use the **Attribute** field to qualify the event subscription; for example, to specify a change from one value to another. For this example, leave the default to subscribe to *any change* in the attribute value (only the **Change** radio button is selected).
8. In the **Event Monitor Action** field, click on both the options **Highlight Event Entry** and **Popup Event Monitor**. This specifies that if the subscribed event occurs, you want the Event Monitor window to be displayed with the event entry highlighted.



Description: My Server 2

Path: larner.hursley.ibm.com<Host Image><larner.hursley.ibm.com><Server Ima

Subscription Type : ● Creation ○ Deletion ○ Relation ● Attribute

Relations / Attributes :

number of method responses
operating mode
is server group
lifecycle registration status
naming registration status
server name
visible in cell name tree
visible in work group name tree
process priority
server group enable

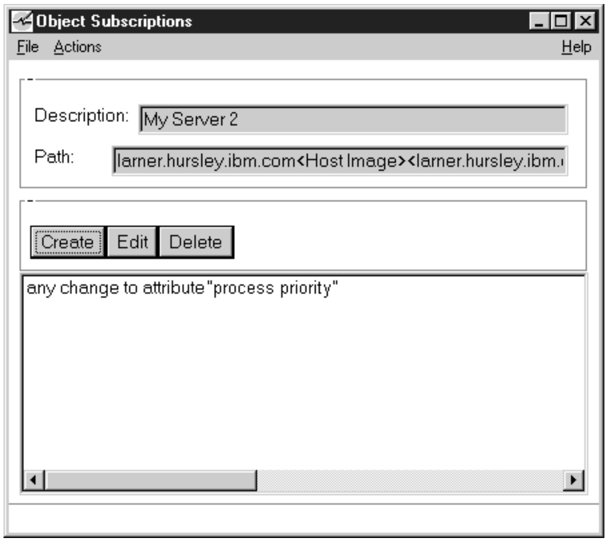
Attribute :

● Change ☐ From : low

● Threshold ☐ To : low

Event Monitor Action : ☑ Highlight Event Entry ☑ Popup Event Monitor

9. Click on the **OK** button, to create the event subscription.



Description: My Server 2

Path: larner.hursley.ibm.com<Host Image><larner.hursley.ibm.

any change to attribute"process priority"

Note that in the Object Subscriptions window there is now an entry:

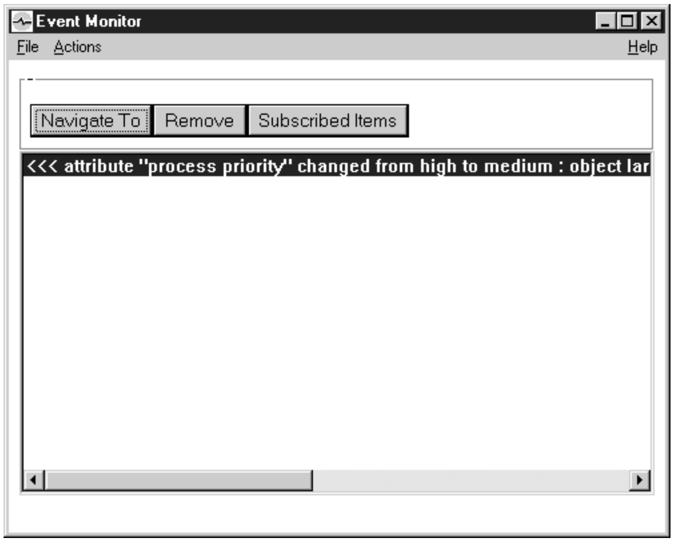
```
any change to attribute "process priority"
```

(If the Object Subscriptions window is not visible, click on the **Object Subscriptions** button of the Windows NT task bar.

To demonstrate what happens when this event occurs, edit the attribute and change its value to *production*, as follows:

1. Click the right mouse button on **My Server 2**, to display its pop-up menu
2. Click on **Edit**, to display the Object Editor window
3. Click on the **Main** tab on the right of the notebook
4. Click on **process priority**
5. Click on the down arrow to the right of the **Attribute Value** field, to display the alternative value **production**
6. Click on **medium**
7. Click on the **OK** button, to apply the change and close the Object Editor window

This causes CBConnector System Management to register the event and display the Event Monitor window with the event message.



<<< attribute "process priority" changed from high to medium : object lar

## Stopping the server called My Server 2

While a server is running, if you click on its Server Image, the Status Bar displays the status of the server. For example, for the server **My Server 2**, the status line displays: **My Server 2: runImmediate running 3 3 good**, where **running** indicates that the server is running.

To stop the server, on the pop-up menu of **My Server 2** click **Stop**.

To confirm that the server has stopped, click on **My Server 2.** When the server is stopped, the status line will display: **My Server 2: runImmediate stopped 0 0 poor**, where **stopped** indicates that the server has stopped.

### Related Tasks

"Edit Objects" on page 72
"Example: Change the Values of Attributes" on page 198
"Example: Configure Applications on Servers" on page 248

# Chapter 8. Administer Clients

You can define the properties of a type of client as a *Client Style*, which can be configured onto any number of hosts to enable them to support clients with that style. All clients *of the same style* on a host share the same configuration information. This information is copied from the Client Style and stored on the host as a *Client Style Image*.

Component Broker provides a default client style, called **defaultClientStyle**. The System Manager automatically creates a Client Style Image called defaultClientStyle on every host that you associate with any Configuration. You can also create your own client styles, giving each a unique name of your choice (typically a name representative of the use of the client style).

When you activate a system management Configuration, each of its Client Style models configured on a Host model are used to produce a corresponding client style properties file, which is then stored at each respective host. Whether or not the properties file is generated, and where the resulting properties file is placed in the host's file system are controlled by a pair of attributes on the Host model. By default, properties files are automatically generated, and they are placed in **c:\CBroker\data\properties** (on Windows NT) or in **/usr/lpp/CBroker/data/properties** (on AIX).

If you want to prevent the properties files from being generated, or if you want them placed in a different directory then you can change the corresponding attributes in the main notebook page of the Host model.

For C++ client applications running on managed client hosts, or C++ client applications running on server hosts, the client process is associated with its client style through the **SOMCBENV** environment variable.

Java clients (both Java applications and applets) get their client style attribute values from Java property files, as "Configured Clients" on page 211. The Java client is associated with its properties file by the URL specified on the command line with which the client was invoked.

With either a Java application client or Java applet client you can place the client style properties file on a Web Server; although this is much more prevalent for Java applet clients. When the client style properties are placed on a Web Server, they can be shared by many clients. For those clients that you want to share the same properties file, you only need to specify the same URL when the client is invoked.

Because you specify a client's keyring in its properties file, all clients sharing the same properties file also use the same keyring. That means that all clients using the same properties file have the same set of trust-basis in the servers they use. It also means that all such clients are subject to the same security and quality of protection policies.

However, you should consider the following trade-offs for clients using the same properties file:

- Having a common client style that is shared over a large number of clients reduces your administrative burden. If you decide to change a property of the client style (for example, its security policy or the keyring), you can change it in one place and effectively change it for all of the clients that are using that client style.

**Note:** A client style's security policy includes the clients' trust-basis, which defines the set of servers whose authenticity you trust based on the fact that the trust-basis exists in the certificate chain that those clients present through SSL.

- Different clients may need different security policies. If this is needed, you must provide a different client style for each combination of security policies that you want to use.
- Having a common Web Server from which you get the client style, keyrings, java classes, and so on represents a single point of failure, depending on the robustness of your Web Server. This may lead you to put several client styles (even if they contain all the same configuration values) on different Web Servers, and then direct different clients to use different Web Servers. Then, if one Web Server goes down it will only affect the clients using that Web Server, leaving other clients to continue their business functions.
- If one of your servers is compromised then you must replace the certificates in the certificate chain up to the trust-basis you have established for your clients. You then must replace the certificates in the client keyrings that use that trust-basis.

  Reducing the number of client styles that refer to the same keyring, and therefore the same trust-basis, reduces the number of keyrings and client styles that need to be updated. This reduces the administrative burden of recovering from a compromise.

  However, reducing the number of client keyrings that refer to the same trust-basis increases the number of clients that are vulnerable to compromises in the trust-basis if that compromise is not detected or reported.

  This trade-off is normally overshadowed by your reasons for needing a common security policy, or more specifically for using the same trust-basis for your clients. If you need for a large number of your clients to use the same set of servers, then this need overshadows any consideration you might give to increasing the number of clients styles and keyrings you use across your clients.

### Related Concepts

"Client Styles" on page 26
"Client Styles" on page 26
Java Application Clients
Java Applet Clients

### Related Tasks

"Configure a new Client Style" on page 151
"Configure Applications onto a Client Style" on page 213
"Define and Configure Client Styles" on page 212
"Add a Client Application into a Configuration of your Application Environment" on page 230

## Clients

The main types of clients supported by Component Broker are as follows:

- “Java Application Clients” on page 208
- “Java Applet Clients” on page 209
- “C++ Clients” on page 210
- “Configured Clients” on page 211

CBConnector System Management can manage clients through an SM agent, which can be used to dynamically change the properties of such clients. If a client is on a separate host, that host has an SM agent, identical to that of a server host.

Alternatively, CBConnector System Management can manage *configured clients* through their client style properties files. The file is read when such a client starts, so the properties of such clients can only be changed when the client is started.

You define the properties of a type of client as a *Client Style*, which can be configured onto any number of hosts to enable those hosts to support clients with that style. For more information about the relationship between clients and client styles, see “Chapter 8. Administer Clients” on page 205.

To define a client style, you can use the **Create Clients** wizard, as described in “Configure a new Client Style” on page 151.

To configure a client style, you can use the **Configure Clients** wizard, as described in “Configure Applications onto a Client Style” on page 213.

System management of managed clients is through the System Manager user interface. Actions are passed from the System Manager to the SM agent controlling the client.

For client-only hosts, the SM agent does not need to be running all the time. On client-only hosts, an SM agent-attached user interface is provided so that the agent can be started and stopped locally. This does not interrupt the operation of client processes or affect their configuration information. Stopping the agent saves resources on the client-only host, but requires the agent to be started manually when system management is needed. An example of where this might be useful is when the client host runs without an agent, to optimize performance, until the user reports a problem. The administrator could then ask the user to start the agent (or do so remotely) and then the administrator can set trace levels, access error logs, and perform actions remotely through the System Manager user interface.

### Related Concepts

“Client Styles” on page 26
“Java Application Clients” on page 208
“Java Applet Clients” on page 209
“C++ Clients” on page 210
“Configured Clients” on page 211

### Related Tasks

“Configure a new Client Style” on page 151
“Configure Applications onto a Client Style” on page 213

“Define and Configure Client Styles” on page 212
“Chapter 8. Administer Clients” on page 205

## Java Application Clients

A *Java Application client* is one where you use one or more client applications written in Java that you have installed at that client workstation. There are actually variations on this as you can copy the application binary files onto a remote workstation and load it in to the memory of your client workstation using a distributed file system. But the key characteristic of the Java Application client is that the Java Virtual Machine (JVM) believes it is loading the application from a local file system; particularly that it *is not* loading it from a remote Web Server.

Java applications are often run within a native JVM; that is, without executing within a Web Browser. For example, you can start a Java application with the following command-line invoke string:

```
java -Dcom.ibm.CORBA.BootstrapHost=bootstrap_host_name
-Dcom.ibm.CORBA.ClientStyleImageURL=URL_of_properties_file
application_name
```

**Where:**

**-Dcom.ibm.CORBA.BootstrapHost=***bootstrap_host_name*
is used to specify the bootstrap host name that should be used by this client.

**-Dcom.ibm.CORBA.ClientStyleImageURL=***URL_of_properties_file*
is used to specify the URL of the client style properties file to be used by this client.

The Java application invokes operations on business objects distributed across application servers on a variety of hosts in the enterprise. The location of these objects is transparent to the application. The business objects on application servers can, and often do, invoke requests on other business objects either locally in the same application server or on any other application server. Again, the location of these business objects are transparent to the requesting business objects.

**Java Application Security:**

JVMs generally operate with two different security models; one that applies to Java Applications and another that applies to Java Applets. For a Java application, the JVM allows the classes of that application to also use other classes from the local file system. This means that you can install the Component Broker Java client runtime at the same client machine; including the Component Broker security client service features. You can also install the DCE client library that Component Broker uses to provide you full 3-party authentication and DCE-enabled message protection.

Even if you plan to use SSL-based authentication at your Java client, the fact that you can access the local file system from within a Java application client also means that you can define and install a Certificate Keyring that is unique to that specific client.

**Related Concepts**

“Java Applet Clients” on page 209

“C++ Clients” on page 210
“Clients” on page 207
“Client Styles” on page 26

**Related Tasks**

“Configure a new Client Style” on page 151
“Configure Applications onto a Client Style” on page 213
“Define and Configure Client Styles” on page 212
“Chapter 8. Administer Clients” on page 205

## Java Applet Clients

A *Java Applet client* is distinguished first because the application is written as a Java applet, with its classes downloaded from a Web Server. Also, a Java Applet can be distinguished by inheriting from the Applet base class.

A Java applet is installed at the Web Server and downloaded, most often, in response to a Java statement in an HTML web document, in the following form:

```
<APPLET CODE = applet_name WIDTH=width HEIGHT=height >
<PARAM NAME=org.omg.CORBA.ORBClass VALUE="com.ibm.CORBA.iiop.ORB" >
<PARAM NAME=com.ibm.CORBA.BootstrapHost VALUE=bootstrap_host >
<PARAM NAME=com.ibm.CORBA.ClientStyleImageURL VALUE=cs_url >
</APPLET>
```

**Where:**

**PARAM NAME=com.ibm.CORBA.BootstrapHost VALUE=***bootstrap_host*
is used to specify the bootstrap host name that should be used by this client.

**PARAM NAME=com.ibm.CORBA.ClientStyleImageURL VALUE=***cs_url*
is used to specify the URL of the client style properties file to be used by this client.

Java applet clients are typically used when trying to avoid having to install any software at the client host. Not only does this make the client footprint smaller, but it avoids having to manage software versions at the client machines. This should significantly reduce the cost of deploying client workstations and enables the Network Computing (NC) model of computing.

If the Java applet needs the Component Broker client runtime to locate the business objects it uses, the Component Broker client runtime is downloaded automatically from the Web Server when the Java applet is downloaded. For more information about installing and configuring the Component Broker Java client on a Web server as a downloaded Java applet, see "Installing the Java Client on a Web Server" in the *Quick Beginnings*.

The Java applet invokes operations on business objects configured on an application server on the Web Server host. Those business objects, in turn can invoke operations on other business objects distributed across application servers on a variety of hosts in the enterprise. The location of these objects is transparent to the business objects on the Web Server host.

Downloading a client application from a Web Server instructs the Java Virtual Machine (JVM) to treat it as an applet, and to enforce the applet security model.

**The Java Applet Security Model:** The applet security model has the following properties, but can be customized by different Web Browser environments:

- If a downloaded Java class has been digitally signed, and if the signer has been authorized (defined as a trust-basis) by the client's Web Browser keyring, the class can make use of any local file system or other local I/O facilities or resources. Also, the downloaded class can establish communication (for example, using a TCP/IP socket) with any other host.
- If a downloaded Java class *has not been digitally signed and authorized*, it can only make use of Java classes, files, and resources that have also been downloaded from the same Web server. Also, it can only establish communication with the Web Server from which it was downloaded.
- If a Java class is found in the classpath of the local client, even if invoked by a downloaded Java class, it is loaded from the local file system instead of being downloaded from the Web Server. At that point, irrespective of whether that class has been digitally signed, that class is treated as an application class and given full access to the local file system or other local I/O facilities or resources. Also, it can establish communication (for example, using a TCP/IP socket) with any other host.

Because Java Applets are downloaded from a Web Server, they are almost always invoked from within a Web Browser; either from another Java application, or more commonly from an HTML Web Document. In the latter case, assuming that you are using HTTP-S, the Java Applet can only be invoked and downloaded from the same Web Server (host) from which the HTML document was downloaded.

Also, a Java applet client can only use the SSL-based authentication model offered by Component Broker. This is because the DCE client library itself is not downloadable; it is not implemented in Java. If you are using a Java Applet client, you presumably would not want to make use of anything that was not also downloaded. However, the SSL protocol stack is implemented in Java as a downloadable applet and, therefore, can be used from an applet client.

### Related Concepts

"Java Application Clients" on page 208
"C++ Clients"
"Clients" on page 207
"Client Styles" on page 26

### Related Tasks

"Configure a new Client Style" on page 151
"Configure Applications onto a Client Style" on page 213
"Define and Configure Client Styles" on page 212
"Chapter 8. Administer Clients" on page 205

## C++ Clients

A *C++ client* is one where you use one or more client applications written in C++ that you have installed at that client workstation.

C++ client applications can be written in either VisualAge C++ or Microsoft's Visual C++. In the latter case the client normally uses ActiveX controls that wrapper the CORBA proxies to the business objects they use. Further, the latter case also applies equally to client applications that are written in Microsoft's Visual Basic.

**Related Concepts**

"Java Application Clients" on page 208
"Java Applet Clients" on page 209
"Clients" on page 207
"Client Styles" on page 26

**Related Tasks**

"Configure a new Client Style" on page 151
"Configure Applications onto a Client Style" on page 213
"Define and Configure Client Styles" on page 212
"Chapter 8. Administer Clients" on page 205

## Configured Clients

A *Configured client* is distinct from a managed client in that it allows centralized control over configuration of some attributes of the client without needing an SM agent running on the client host. This is particularly useful for downloadable Java applet clients where we cannot expect a local SM agent to exercise control over such client processes.

The idea of a configured client currently only applies to Java clients and for a subset of configuration attributes; particularly security attributes. A configured client works by reading its configuration attributes from a particular Client Style Image. If the client resides on a managed host, then it reads the Client Style Image from the SM agent on that host. However, normally the client is on a non-managed client host machine and reads its Client Style Image from a remote host. In Java, this is done by copying the Client Style Image values to a Java properties file either on the local client host or, more often, on a Web Server accessible from the client. The Client style properties file is then read when the client starts.

Any managed host can be configured with a number of client styles, each with a different name. The System Manager generates the corresponding client style properties file for every Client Style model in a Configuration when you activate that configuration. When you activate a Configuration containing a configured Client Style, the System Manager creates the Client Style Image on the hosts on which the Client Styles are configured, and produces the corresponding client styles properties files at each of those hosts. It is then up to you to copy the client style properties file to somewhere that your configured clients can locate them.

When you start a Java client you must tell it the name of the client style properties file to use. You specify the URL for the client style properties file as a property argument on the Java command line. This property value is then used by the ORB during its initialization to locate and read the corresponding client style properties file specified at that URL. If your configured client is to run as a downloadable Java applet, then you should copy the client style properties file to your Web Server and set the URL on the Java command line to refer to that client style properties file at your Web Server. If your configured client is to run as a Java application, then you can refer to a client style properties file on the client's host.

Because the client style properties file is only read when the client starts it cannot be used to change the configuration of the client process dynamically as you might expect with a managed client. Instead, any configuration changes are recorded in the client style properties file and read the next time the client starts again.

**Related Concepts**

Java Application Clients
Java Applet Clients
“C++ Clients” on page 210
“Clients” on page 207
“Client Styles” on page 26

**Related Tasks**

“Configure a new Client Style” on page 151
“Configure Applications onto a Client Style” on page 213
“Define and Configure Client Styles”
“Chapter 8. Administer Clients” on page 205

## Define and Configure Client Styles

To use clients within your enterprise, you have to define and configure the style of clients within a system management Configuration. When you next activate the Configuration, the System Manager enables those client styles on their hosts.

The style of a client, defined by a **Client Style** model determines the attributes of its clients and the applications that are to run on them. The Client Application models configured on a Client Style model define the client applications to be used on that style of client.

To define a client style, you can use the **Create Clients** wizard, as described in “Configure a new Client Style” on page 151.

To configure a client style, you can use the **Configure Clients** wizard, as described in “Configure Applications onto a Client Style” on page 213.

Alternatively, you can use the following sequence of subtasks to create and configure a client style:

1. Create a Client Style model (if it does not already exist)
2. Configure the Client Style model as required

**Related Concepts**

"Client Platforms and Configurations" in the *Quick Beginnings* book
“Client Styles” on page 26
“Clients” on page 207
“Wizards” on page 447

**Related Tasks**

“Configure a new Client Style” on page 151
“Configure Applications onto a Client Style” on page 213
“Configure Security for a Client Style” on page 326
“Configure a Client Style to use Session Services” on page 411
“Disable the ORB’s Enhanced Workload Management (WLM) Extension” on page 216
“Configure a Client Style onto a Host” on page 217
“Add a Client Application into a Configuration of your Application Environment” on page 230

page 230
"Configure Applications onto a Client Style"

## Configure Applications onto a Client Style

To configure applications onto a client style within a selected Configuration of your enterprise, you can use the **Configure Clients** wizard, as follows:

1. On the menu-bar of the System Manager user interface, click **Wizards - Configure Client**.
2. On the Select Applications to Configure panel, select the applications that you want to configure onto the server group.

   To add an application to the list to be configured, displayed under *Applications to Configure*, complete the following steps:

   a. In the *Available Applications* list box, click on one or more applications. To select several applications, press and hold the Ctrl key while clicking on the applications then release the Ctrl key.

   b. Click the **Add** button.

   The applications are added to the list of applications to be configured on the client style, listed in the *Applications to Configure* list box.

   To remove an application from the *Applications to Configure* list box, complete the following steps:

   a. Select the application's entry in the *Applications to Configure* list box.

   b. Click the **Remove** button.

   The application is removed from the list of applications to be configured on the client style.
3. On the Management Zone panel, select the application Management Zone within which the client style is defined.
4. On the Configuration panel, select the Configuration (of the Management Zone that you selected previously) within which the client style is defined.
5. On the Client Style panel, select the name of one or more client styles onto which the applications are to be configured.

   To add a client style to the list under *Client Styles To Configure Applications On*, complete the following steps:

   a. In the *Available Client Styles* list box, click on one or more client styles. To select several client styles, press and hold the Ctrl key while clicking on the client styles then release the Ctrl key.

   b. Click the **Add** button.

   The client styles are added to the list of client styles in the *Client Styles To Configure Applications On* list box.

   To remove a client style from the *Client Styles To Configure Applications On* list box, complete the following steps:

   a. Select the client style's entry in the *Client Styles To Configure Applications On* list box.

   b. Click the **Remove** button.

   The client style is removed from the list of client styles onto which the selected applications are to be configured.

6. To finish and have the wizard configure the applications onto the client style, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

   If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.

**Note:** The wizard does not change the client style until you click the **finish** button on any panel and have specified the details that it needs. This updates the client style within the Configuration specified, but does not change the clients that use that client style in your enterprise.

The clients in your enterprise are changed the next time that you activate the Configuration.

**Related Concepts**

"Client Platforms and Configurations" in the *Quick Beginnings* book
“Client Styles” on page 26
“Clients” on page 207
“Wizards” on page 447

**Related Tasks**

“Configure a new Client Style” on page 151

## Configure the Request Timeout for Client Styles

Use this procedure to increase the request timeout for all clients of an application server that is set to run on request.

The default request time out value for client styles is too short when an application server’s **run control** attribute is set to `run on request`. This is because a run on request application server must be started when it receives its first request. The response time for this first request is therefore much longer than the response time for subsequent requests.

Therefore, if an application server’s **run control** is set to `run on request`, then you should increase the value of the **request timeout** attribute for all clients of that server.

If the request timeout for a client is too short, you would see error log entries like those shown in error log entry for a client process (page 215) at the end of this topic:

**To change the request timeout for the Client Styles, complete the following steps:**

1. Start the System Manager User Interface, and return to the home view; for example, by clicking the 🏠 icon of the Tool bar.
2. Expand the Management Zones folder
3. Expand the Management Zone within which the application server is defined
4. Expand the Configuration within which the application server is defined

5. Expand the Client Styles folder
6. From the pop-up menu of the Client Style, select **Edit**.
7. Select the **Orb** notebook page tab.
8. Type an appropriate value in the field for **Request timeout** attribute
9. To apply the change, and close the Object Editor window, click **OK**.
10. Repeat steps 5 through 8 for each Client Style that uses the run on request application server.
11. To check that Configuration is valid and update the request timeout for all the client styles, activate the Configuration.

    On the pop-up menu of the Configuration, click **Activate**.

    The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

### Error log entry for a client process

When a client process’s request time out is too short an error log entry like following is created:

```
————————————————————————————————-
ComponentId:    393316
ProcessId:      327
ThreadId:       178
FunctionName:   CallStreamIIOP::get_response(::Request&)
ProbeId:        836
SourceId:       1.20 src/orb/src/somd/somderr.cpp
Manufacturer:   IBM
Product:        Component Broker
Version:        2.0
SOMProcessType: 1
ServerName:
clientHostName:
clientUserId:
TimeStamp:      10/29/98 15:06:25.679151835
UnitOfWork:     19678:dwjnt2
Severity:       1
Category:       2
FormatWarning: 0
PrimaryMessage: The function CallStreamIIOP::get_response(::Request&):836 reported an activity.
ExtendedMessage:
A SystemException occurred: NO_RESPONSE, minor code 1229062205 (SOMDERROR_CommTimeOut) at
                                         CallStreamIIOP::get_response(::Request&) line 836.
RawDataLen:     0
——————————————-
```

SOMProcessType 1 indicates the client and the ExtendedMessage indicates the timout.

**Related Concepts**

“Controlled Server Groups” on page 336
“Workload Management (WLM)-Enhanced clients” on page 338

**Related Tasks**

“Configure how Application Servers are to Run” on page 187

## Disable the ORB’s Enhanced Workload Management (WLM) Extension

This topic shows how to use the System Manager User Interface to configure a Client Style Model to ensure that instances of clients created from that model do not enable the enhanced workload management-extension of the ORB.

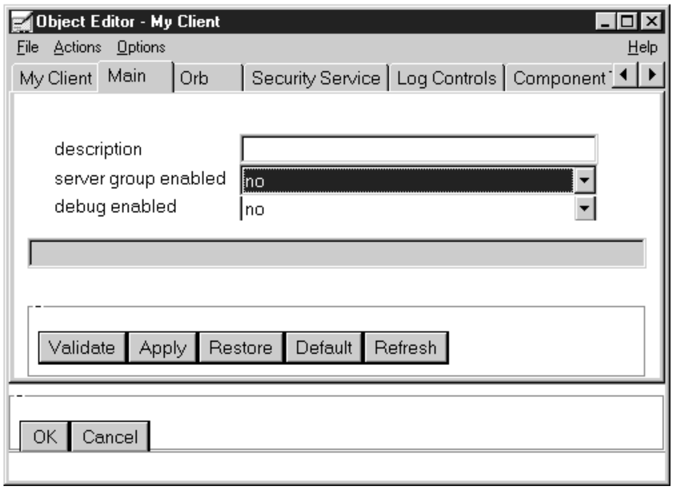
Disabling this ORB extension prevents the DLL (or shared library) files associated with workload management from being loaded by the client. Not loading these files improves the performance of client initialization.

Before carrying out this task, you should make sure you are familiar with:

- Controlled Server Groups
- Workload Management (WLM)-Enhanced Clients

To disable the ORB’s enhanced WLM extension for a Client Style, complete the following steps:

1. Start the System Manager user interface.
2. Open the required Configuration of one of your application Management Zones.
3. Open the Client Styles folder and select the required Client Style.
4. From the pop-up menu of the Client Style, select **Edit**.
5. Select the **Main** notebook page tab.
6. Select the **server group enabled** attribute and change the value to **no**, as shown in Figure 44. (*The snapshot was taken from the Windows NT version of Component Broker.*)



description

server group enabled no

debug enabled no

*Figure 44. The Object Editor for a Client Style*

7. To apply the change, and close the Object Editor window, click **OK**.

The change will only become effective when you next activate the Configuration that contains the Client Style.

**Related Concepts**

“Controlled Server Groups” on page 336
“Workload Management (WLM)-Enhanced clients” on page 338

**Related Tasks**

“Configure a Controlled Server Group” on page 339

## Configure a Client Style onto a Host

Use this procedure to configure a Client Style onto a Host, to specify that the style of client is to be supported on that host.

When you first configure a new client style, you specify a host on which it is to supported, and the bootstrap host for the client style, as described in “Configure a new Client Style” on page 151.

You only need to use this procedure if you want a client style to be supported on other hosts. When the Configuration is next activated, the client style (and all the applications configured on it) will be supported on the new hosts.

1. Expand the Configuration of the application Management Zone within which the Client Style is defined
2. Expand the Client Styles folder.
3. On the pop-up menu of the Client Style, click **Drag**.
4. Expand the Hosts folder
5. On the pop-up menu of the Host that is to support the client style, click **Configure Client Style**. This creates a *Configured Client Styles* relationship between the Client Style and the Host that it has been configured on.

   Repeat this step for each host that you want to support the client style.
6. Optionally, to verify that the relationship has been created, expand the Host. You should see a *Configured Client Styles* relationship with the name of the configured Client Style (from 3). If not, repeat steps 3 to 4.

**To check that Configuration is valid and make the client style available on the new hosts, activate the Configuration.**

On the pop-up menu of the Configuration, click **Activate**.

The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

**Related Concepts**

“Client Styles” on page 26
“Clients” on page 207

**Related Tasks**

“Configure a new Client Style” on page 151

“Configure Applications onto a Client Style” on page 213
“Change the Bootstrap Host for a Client Style”
“Define and Configure Client Styles” on page 212

## Change the Bootstrap Host for a Client Style

Use this procedure to change the bootstrap host for a Client Style. That host provides the local root naming context for objects that cannot be resolved on the host on which the client style is configured.

Client-only machines (those without any server processes) do not have a host name tree. They use the name tree root of another host, referred to as the *bootstrap host*, from which the client is bootstrapped in to the distributed system. The host name tree, in effect, ends up being shared by all the client machines that identified that host as their bootstrap host.

When you first configure a new client style, you specify its bootstrap host, as described in “Configure a new Client Style” on page 151.

You only need to use this procedure if you want a client style to use a different bootstrap host. When the Configuration is next activated, the client style will use the new host to be bootstrapped into your enterprise.

**To change the bootstrap host for a Client Style, complete the following steps:**

1. Expand the Configuration of the application Management Zone within which the Client Style is defined.
2. Expand the Client Styles folder.
3. Expand the Client Style that you want to change.
4. Delete the current Host for Bootstrap relationship for the Client Style. To do this, complete the following steps:
   a. Expand the Host for Bootstrap folder.
   b. From the pop-up menu of the Host in the Host for Bootstrap folder, click **Delete**. This displays a dialog box for you to confirm that you want to delete the Host for Bootstrap relationship for the Client Style.
   c. In the Deletion dialog box, click the **Yes** button.
5. On the pop-up menu of the Client Style, click **Drag**.
6. Expand the Hosts folder
7. On the pop-up menu of the Host that is to provide the bootstrap service, click **Configure Client Style for Bootstrap**. This creates a *Host for Bootstrap* relationship between the Client Style and the Host that it has been configured on.
8. Optionally, to verify that the relationship has been created, expand the Host for Bootstrap folder. You should see a Host object with the name from 7. If not, repeat steps 5 to 7.

**To check that the Configuration is valid and make the client style use the new bootstrap host, activate the Configuration.**

On the pop-up menu of the Configuration, click **Activate**.

The System Manager displays an Action Console window that you can use to monitor the progress of the **Activate** action. The console first displays messages about the System Manager verifying that the Configuration is valid. If you have completed the above steps properly, you should see a *Configuration valid* message. The console then displays messages for activating parts of the Configuration. Finally, if you have completed the above steps properly, you should see a *Activation successful* message.

#### Related Concepts

"Client Styles" on page 26
"Clients" on page 207

#### Related Tasks

"Configure a new Client Style" on page 151
"Configure Applications onto a Client Style" on page 213
"Configure a Client Style onto a Host" on page 217
"Define and Configure Client Styles" on page 212

## Log in a Client Principal

Use this procedure to log in a client principal that is to use Component Broker.

Servers, by default, log in automatically with the information in their keytab file.

**Prerequisite:** You must have created a DCE account for the client principal, as described in "Create an Account for a Client Principal" on page 317

You can log in a client principal in a variety of ways, depending on what client and security mechanism you are using.

- To log in explicitly from a CORBA C++, ActiveX, or Java Application client that is enabled to use the DCE security mechanism, type the following **dce_login** command at a command prompt:

  ```
  dce_login <security name>
  ```

  Where <security name> is the security name for your client principal, established by your DCE account. You will then be prompted for your password.

  If you do not log in explicitly, Component Broker will automatically prompt you for your userid and password at runtime when it attempts to form a secure association for you from your client to an application server.
- If you are using a Java applet or application client that has been enabled for SSL-based authentication, then Component Broker will automatically prompt you for your userid and password at runtime, when it attempts to form a secure association for you from your client to an application server. (From an SSL-based Java applet or Java application client, you can not use the dce_login command to log in.)

To enable Component Broker to automatically prompt you for your userid and password, you must have set the **login source** attribute to **prompt** in the Client Style model for that client, as described in "Configure Security for a Client Style" on page 326.

#### Related Concepts

"SSL and Certificates" on page 277
"Certificate Authorities" on page 280

“Accounts for Component Broker Administration” on page 287
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

“Create an Account for a Client Principal” on page 317
“Change the Password for a Client Principal” on page 319
“Administer Accounts for Client and Server Principals” on page 316
“Configure Security for a Client Style” on page 326
“Chapter 11. Administer Security in your Enterprise” on page 261

# Chapter 9. Administer Applications

This topic provides information about administering applications in an existing application environment.

If you have not yet configured your application environment, you should see "Chapter 5. Configure a new Application Environment" on page 133.

If you want to administer the hosts on which the application environment runs, see "Chapter 6. Administer your Host Environment" on page 167.

The overall procedure for administering an application is as follows:

1. "Installing Applications" on page 222, typically onto the System Manager host. This enables the System Manager to distribute the application to the hosts on which it is to run.
2. "Define and Configure Applications" on page 227. This enables you to tailor the attributes of the application and to relate it to the server groups or freestanding servers on which it is to run.
3. Distribute the application to the hosts on which it is to run. When you activate a system management Configuration that defines your application environment, the System Manager automatcally distributes the applications to the hosts on which they are to run.
4. Optionally, Refresh an application (page 223). When you receive replacement application software, you can update the applications configured within Component Broker by simply refreshing the applications.
5. Optionally, Uninstall an application (page 223). When you no longer want an application in your Component Broker application environment, this enables you to remove the application.

Besides the tasks listed in the above topics, see the related tasks for others that impact you application environment. For example, "Chapter 13. Administer Connections to Tier-3 Systems" on page 343 contains tasks about configuring Connections that Component Broker applications use to communicate with applications running on tier-3 systems.

**Related Concepts**

"Applications" on page 27

**Related Tasks**

"Chapter 5. Configure a new Application Environment" on page 133
"Chapter 7. Administer Application Servers" on page 181
"Chapter 8. Administer Clients" on page 205
"Chapter 11. Administer Security in your Enterprise" on page 261
"Chapter 13. Administer Connections to Tier-3 Systems" on page 343
"Chapter 14. Administer Component Broker Services" on page 397
"Start an Application" on page 244
"Stop An Application" on page 244

## Installing Applications

Each CBConnector application comprises a number of files provided in a unique directory. This collection of files is known as an *application family*. One of the files, called *familyname.DDL*, is used to tell the System Manager what system management objects are to represent and manage the application. For example, the DDL file identifies Application Install and DLL Install objects needed.

The general subject of installing applications, described in this topic has two related subjects: Uninstalling applications (page 223) and Refreshing applications (page 223), outlined at the end of this topic.

Each CBConnector application normally provides its own installation tool, with its own user interface. The tool provides actions to install, uninstall, and refresh the application.

**Installing new applications:**

When a new application is installed into Component Broker, the install action copies the application files into a unique directory for the application family.

For an application that is to run on a server, its installation tool creates an *Application Family Install* object to represent the application family, and creates other Install objects to represent the applications, DLLs, classes, and other components. It also adds the application into the *Available Applications* folder that is displayed on the Home view of an Information Controller window. This folder contains objects for all the server applications that have been installed onto the System Manager's host.

For an application that is to run on a client, its installation tool creates a *Client Application Family Install* object to represent the application family, and creates other Install objects to represent the applications, DLLs, classes, and other components. It also adds the application into the *Available Applications* folder that is displayed on the Home view of an Information Controller window. This folder contains objects for all the client applications that have been installed onto the System Manager's host.

You must install an application onto the System Manager's host, so that it is aware of the available application. The System Manager can later distribute automatically the application to other hosts according to the configurations that you create. This makes it easier for you to manage the installation, distribution, and uninstallation (if needed) of applications in a multi-host enterprise.

If you are starting with a single host, and later expand to a multi-host enterprise you can benefit from this automatic distribution of applications to the new hosts.

(You can install an application directly onto another host on which it is to be run, but normally you do not do this because the System Manager will download the application to that host if needed.)

The System Manager distributes an application automatically when its Configuration is activated. The application files are copied to the hosts on which the application is to run. This is determined by the following relationships that you define in a Configuration:

- The servers (or client styles) that the application is to run on

- The hosts on which the servers (or client styles) are to run

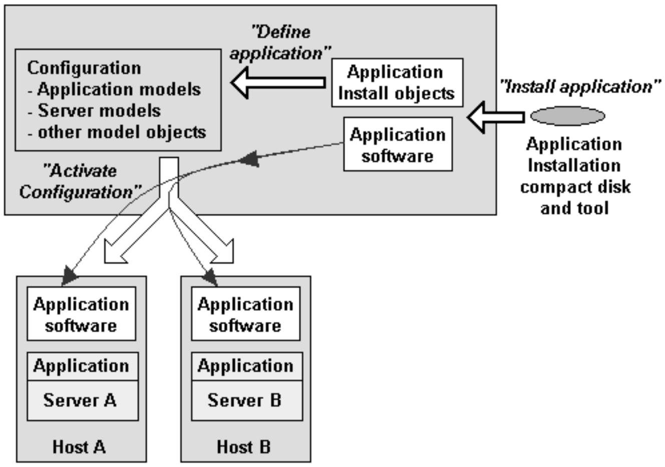
This is outlined in the figure Copying of Application Files (page 223).

**Note:** If an application needs the Interface Repository (IR) to be populated on target hosts to which it has been distributed by the System Manager, you must do so manually. To update the IR for an application after it has been distributed, complete the following steps on each target host:

1. Copy the IR update utility provided with the application to each target host from the System Manager host
2. Run the IR update utility on the target host

For the name of the IR update utility, and instructions about using it, see the information provided with the application.

**Copying of Application Files by an Application Installation Tool and subsequently by the System Manager when Activating a Configuration**



The instructions for installing an application depend on the application and the installation tool that it uses. Therefore, to install an application normally, see the tool and instructions provided with the application.

Alternatively, you can install an application manually using the **Load Application** action, but this is normally only needed in an application development environment.

**Uninstalling applications:**

Uninstalling applications from Component Broker is the reverse of installing applications. As with installing, you normally uninstall applications from the System Manager's host. Normally, the application installation tool provides an **Uninstall** action that you use to remove the application. You can also use the **Uninstall Family** action through the System Manager user interface, but this is normally reserved for application development environments.

**Refreshing applications:**

If an application is changed after it has been installed, you can refresh the application to update its files and system management objects. As with installing, you normally refresh applications from the System Manager's host. Normally, the application installation tool provides a **Refresh** action that you use to update the application. You can also use the **Refresh** action through the System Manager user interface, but this is normally reserved for application development environments.

### Related Concepts

"The Install World" on page 41
"Applications" on page 27

### Related Tasks

"Load a new Application" on page 141
"Add an Application into a Configuration of your Application Environment" on page 228
"Uninstall an Application"
"Refresh an Application" on page 225

## Uninstall an Application

Use this procedure to uninstall applications from Component Broker System Management. *This procedure is normally used only in an application development environment before a proper tool is available to uninstall an application.*

Normally, this procedure is completed automatically by the **uninstall** action of the application's installation tool. For instructions and guidance about this, see the information provided with the application's installation tool.

This procedure can be completed using the **Uninstall Family** action through the System Manager user interface, as described in this topic. This action, on an Application Family Install object, removes all system management objects for the Application Family. It *does not remove* the application files for the Application Family, in case they contain data that you need to preserve.

If this procedure is performed through the Host Image for the System Manager host, the system management objects for the application family are removed from *all* managed hosts (including the System Manager host). For example, this would remove Application models and DLL models from Configurations, Application Images from Server Images, and the Application Family Install object and all its other Install objects from all Host Images.

You can use this procedure through the Host Image for one managed host to remove the system management objects for the application family from that host only. For example, this would remove Application Images from Server Images, and the Application Family Install object and all its other Install objects, from only that one Host Image. It would not affect any model objects created for the Application Family.

If you only want to stop an application from being used on selected hosts, you need only change your Configuration so that the application is no longer configured to run on those hosts. For example, by deleting an Application model or its *Configured Server Groups* relationship to the model for a server group that the application is to run on. When you next activate the Configuration, the System Manager removes the application from all affected servers or client styles (on one or more managed hosts).

If you want to remove the application files from a host, you can use the normal host commands and utilities to do so, or can use actions on the File System objects through the System Manager user interface. The application files are contained in a directory with the same name as the Application Family Install object.

**To uninstall an application using the Uninstall Family action, complete the following steps:**

1. Start the System Manager User Interface and set the User Level to Expert. (Click **View - User Level - Expert** .)
2. Expand the Host Images folder, then the Host Image from which the application is to be uninstalled.
3. Expand the Application Family Installs folder, to display the Application Family Install icon for the application to be uninstalled.
4. From the pop-up menu of the Application Family Install, click **Uninstall Family**.

You can monitor the progress of the **Uninstall Family** action in its Action Console window. If the action completes successfully, the Action Console window displays a success statement. (If the action fails, the Action Console window displays messages to indicate the cause of the problem.)

Due to the number of objects that need to be accessed, updated, and deleted by this operation, you may see servers cycle several times. If a server fails and does not restart, try “Stop a Server” on page 193. (On the pop-up menu of the Server Image, click **Run** or **Run Immediate** as required.)

When the **Uninstall** action has completed successfully, reset the User Level to your normal level.

**Related Concepts**

“Installing Applications” on page 222
“The Install World” on page 41
“Applications” on page 27

**Related Tasks**

“Load a new Application” on page 141
“Add an Application into a Configuration of your Application Environment” on page 228
“Add a Client Application into a Configuration of your Application Environment” on page 230
“Configure an Application onto a Freestanding Server” on page 186
“Configure Applications onto a Client Style” on page 213
“Refresh an Application”

## Refresh an Application

Use this procedure to refresh (update) an application that has already been installed into Component Broker System Management. This causes the System Manager to automatically update the files used by the application and the system managament objects used to manage the application.

This procedure changes the existing application only as required; for example, it does not change the relationships of a related Application model unless instructed by the DDL file used to refresh the application. This means that you can continue to use your existing system management configurations for the application.

To update the application files, the System Manager automatically completes the following steps:

1. Shuts down all servers on which the application is configured
2. Copies the files to the hosts on which the servers were running
3. Restarts the servers

Normally, this procedure is completed automatically by the application installation tool. However, the procedure can be completed using the **Refresh** action through the System Manager user interface, as described in this topic. *This procedure would normally be used in an application development environment before a proper tool is available to uninstall an application.*

**Before starting this procedure, consider the following points:**

- You refresh an application on the System Manager's host, so that it is updated consistently on all hosts on which the application is run. The System Manager distributes automatically the files and any changed system management objects to those hosts.
- You will need to type the name of the application's DDL file. The System Manager uses the application DDL file to identify the system management objects to be created for the application. The name of the DDL file is also used to identify the directory, *cbinstall/apps/familyname*, into which the application files are to be copied, where:
  - *cbinstall* is the name of the directory into which Component Broker was installed
  - *apps* is the name of the application installation directory for the platform on which the application is to run. Applications to run on Windows NT are copied into *cbinstall/NTApps/familyname*. Applications to run on AIX are copied into *cbinstall/AIXApps/familyname*.
  - *familyname* is the name of the application DDL file

**To refresh an application using the Refresh action, complete the following steps:**

1. Start the System Manager User Interface and set the User Level to Expert. (Click **View - User Level - Expert** .)
2. Expand the Host Images folder, to display the icons for managed hosts.
3. Expand the Host Image for the System Manager's host, to display the folder of Application Families installed on that host.

   (If you are unsure which icon is for the System Manager's host, the name of the System Manager is displayed on the title bar of the Information Controller window.)
4. Expand the Application Family Installs folder.
5. From the pop-up menu of the Application Family Install for which you want to refresh the application, click **Refresh**
6. In the entry field of the *Refresh Application* dialog window, type the full path name of the application's new DDL file.

   You can use the **Browse** button to search for and select the application's DDL file.

   **Notes:**

   a. The Browse dialog box lists directories visible on the workstation ( WIN ) or host ( AIX ) on which the System Manager user interface is running, regardless of which Host Image you select the action on.
   b. ( WIN ) To load a DDL file on a remote host (for example, if the System Manager is running on a remote host), you can map the network drive into which the application was installed on that remote host. Make sure that the account you use to map the network drive has the same permissions as the account under which you are running the CBConnector service.
7. The Refresh Application action copies the application files to the application family directory and reconfigures the application for Component Broker System Management.

You can monitor the progress of the **Refresh Application** action in its Action Console window. If the action completes successfully, the Action Console window displays a success statement. (If the action fails, the Action Console window displays messages to indicate the cause of the problem.)

Due to the number of objects that need to be accessed, updated, and deleted by this operation, you may see servers cycle several times. If a server fails and does not restart, try “Start a Server” on page 192. (On the pop-up menu of the Server Image, click **Run** or **Run Immediate** as required.)

When the **Refresh Application** action has completed successfully, reset the User Level to your normal level.

This procedure has updated the files and system management objects needed by the application.

**Related Concepts**

“Installing Applications” on page 222
“The Install World” on page 41
“Applications” on page 27

**Related Tasks**

“Load a new Application” on page 141
“Define and Configure Applications”
Define and Configure Client Applications
“Uninstall an Application” on page 224

## Define and Configure Applications

After you have “Load a new Application” on page 141, you complete the following general steps to enable the application to be used:

1. “Add an Application into a Configuration of your Application Environment” on page 228 of your application environment. This enables you tailor the attributes of the application and to configure it onto server groups, freestanding servers, or client styles (as appropriate).
2. If required, tailor the attributes and resources for the application.

   Applications are often provided preconfigured, and therefore you only need to add the application to a system management Configuration and configure it onto the servers or client styles that it is to run on. Some applications need special tailoring, for which appropriate information is provided with the application package. See also the information provided in the related tasks below.

3. Configure the application onto server groups, freestanding servers, or client styles, as described in the following topics:
   - “Configure Applications onto a Server Group” on page 183
   - “Configure an Application onto a Freestanding Server” on page 186
   - “Configure Applications onto a Client Style” on page 213

When you next activate the Configuration, the System Manager distributes the applications to the required managed hosts and enables the applications to be used on the server groups, freestanding servers, or client styles on those hosts.

**Note:** When you next activate the Configuration, the System Manager automatically distributes the applications to the required server hosts. If an application needs the Interface Repository (IR) to be populated on those target hosts, you must do so manually. To update the IR for an application after it has been distributed, complete the following steps on each target host:

1. Copy the IR update utility provided with the application to each target host from the System Manager host
2. Run the IR update utility on the target host

For the name of the IR update utility, and instructions about using it, see the information provided with the application.

### Related Concepts

“Applications” on page 27
“Application Servers and Server Groups” on page 25
“Installing Applications” on page 222

### Related Tasks

“Configure an Event Channel for use by Applications” on page 410
“Configure a Single Location” on page 231
“Configure a Chain of Location Objects” on page 234
“Configure a Factory Finder” on page 236
“Chapter 5. Configure a new Application Environment” on page 133
“Chapter 7. Administer Application Servers” on page 181
“Chapter 8. Administer Clients” on page 205
“Chapter 11. Administer Security in your Enterprise” on page 261
“Chapter 13. Administer Connections to Tier-3 Systems” on page 343
“Chapter 14. Administer Component Broker Services” on page 397

## Add an Application into a Configuration of your Application Environment

Use this procedure to add an Application into a Configuration of your application environment so that the application can be configured onto one or more server groups or freestanding servers (as appropriate).

**Prerequisites:** This procedure assumes that you have already loaded the application into Component Broker. If you want to load the application, see “Load a new Application” on page 141, which includes steps to add an application into a Configuration.

This procedure creates an Application in the Configuration by dragging one of the Available Applications and dropping it onto a Configuration. (The application object in the Available Applications folder would have been created when the application was loaded.) This automatically defines the Application with the attributes and relationships that it needs, and creates other system managemement objects needed by the application (in particular; Database Aliases, but also Containers, Policy Groups, C++ Classes, and so on).

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Expand the Available Applications folder
3. On the pop-up menu of the Available Application object, click **Drag**
4. Expand the Management Zones folder
5. Expand your application Management Zone.
6. Expand the Configurations folder of your Management Zone.
7. On the pop-up menu of the Configuration, click **Add Application**
8. Expand the Configuration
9. Optionally, to verify that the Application has been defined, expand the Applications folder. You should see an object with the same name as the Available Application object selected in 3. If not, repeat steps 3 to 7.
10. Optionally, to display the model objects created for the application, expand the Application and its *provides ...* relationship folders.

After you have defined an Application, you probably next want to do one or more of the following:

- Edit the Application to modify its attributes, as described in information provided with the application. (See also “Edit Objects” on page 72.)
- Edit other objects as needed by the application. For information about editing objects for an application, see the information provided by the application.
- Configure the Application onto one or more Server Groups or Servers (free standing), as described in “Configure Applications onto a Server Group” on page 183 or “Configure an Application onto a Freestanding Server” on page 186.

**Note:** Several applications can be processed at the same time, by the following steps:

1. Display a new Information Controller window
2. Open the Available Applications folder in icon view
3. Select several applications.
4. Click **Selected - Drag**
5. On the pop-up menu of the Configuration, click **Add to configuration**
6. Optionally, to verify that the Application models have been created, expand the Application models folder. You should see objects with the same names as the Available Application objects selected in 3. If not, repeat steps 3 to 5.
7. Optionally, to display other objects created for an application, expand the Application model and its *provides ...* relationship folders.

**Related Concepts**

"Application Servers and Server Groups" on page 25
"Applications" on page 27

**Related Tasks**

"Edit Objects" on page 72
"Configure Applications onto a Server Group" on page 183
"Configure an Application onto a Freestanding Server" on page 186
"Add a Client Application into a Configuration of your Application Environment"

## Add a Client Application into a Configuration of your Application Environment

Use this procedure add a client application to a Configuration of your application environment, for use on one or more styles of client defined in that Configuration. To define the application, this procedure creates a Client Application in the Configuration.

**Prerequisites:** This procedure assumes that you have already loaded the application into Component Broker. If you want to load the application, see "Load a new Application" on page 141, which includes steps to add an application into a Configuration.

The Client Application is created by dragging a client application from the Available Client Applications folder and dropping it onto a Configuration. (The client application object in the Available Client Applications folder would have been created when a client application was installed.) This automatically creates the Client Application with the attributes and relationships needed by the application, and creates other objects needed to administer the application (in particular; Database Aliases, but also models for Containers, Policy Groups, C++ Classes, and so on).

To add a client application to a Configuration, complete the following steps:

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Expand the Available Client Applications folder
3. On the pop-up menu of the Available Client Application object, click **Drag**
4. Display the Configuration in which you want the Client Application to be created
5. On the pop-up menu of the Configuration, click **Add Client Application**
6. Optionally, to verify that the Client Application has been created, expand the Client Applications folder. You should see an object with the same name as the Available Client Application object selected in 3. If not, repeat steps 3 to 5.
7. Optionally, to display the other objects created for the application, expand the Client Application and its *provides ...* relationship folders.

After you have defined an Client Application, you probably next want to do one or more of the following:

- Edit the Client Application to modify its attributes, according to information provided with the application. (See also "Edit Objects" on page 72.)
- Edit other objects as needed by the application. For information about editing objects for a client application, see the information provided by the application.

- Configure the Client Application onto one or more Client Styles, so that the application can be run on those styles of clients, as described in “Configure Applications onto a Client Style” on page 213.

**Note:** Several applications can be processed at the same time, by the following steps:

1. Display a new Information Controller window
2. Open the Available Client Applications folder in list view
3. Select several applications.
4. Click **Selected - Drag**
5. On the pop-up menu of the Configuration, click **Add Client Application**
6. Optionally, to verify that the Client Applications have been created, expand the Client Applications folder. You should see objects with the same names as the Available Client Application objects selected in 3. If not, repeat steps 3 to 5.
7. Optionally, to display other objects created for a client application, expand the Client Applicationand its *provides ...* relationship folders.

**Related Concepts**

“Client Styles” on page 26
“Clients” on page 207
“Applications” on page 27

**Related Tasks**

“Configure a new Client Style” on page 151

## Configure a Single Location

Use this procedure to configure a new single location.

You need to use this procedure only if your application program does not create an appropriate single location.

**To configure a single location, complete the following steps:**

1. Display the Configuration that contains the applications that are to use the single location.
2. Create a Single Location model, to be used to define the scope of the single location. To do this, on the pop-up menu of the Configuration, click **New - Single Location**.
3. Configure the Single Location model to define its scope. To do this, complete the following steps:
    a. On the pop-up menu of the Single Location model, click **Edit** to display the Object Editor window
    b. Define the scope of the location, by changing appropriate **scope** attributes from the following list:

        **remote host modifier**
        The transport modifier for the remote host in the form *iiop://host.ip.addr:port*; where *host.ip.addr* specifies the IP address of the remote host, and *port* specifies the numeric IP port that the remote host expects to use.

**remote name context**

The string form of the name needed to resolve to the remote name context from the local root of the remote host; for example, `.:/workgroups/cbworkstations/hosts`, where the name tree for the local host is to be bound as a member host into the work group named `cbworkstations` on the remote host.

**local name context**

The string form of the name needed to resolve to the local name context from the local root; for example, `host`, where the host name tree for the local host is to be bound as a member *host* into the remote name context on the remote host.

**binding direction**

Whether the local name context is bound into the remote name context, or the other way around; default, **`local into remote`**.

**`local into remote`**

The local name context is bound into the remote name context. This option should not be used when the remote name context is part of a Component Broker Windows NT or AIX host environment. In this case, you need to define the relationship in the other network's System Manager and use **remote into local**.

**`remote into local`**

The remote name context is bound into the local name context

**binding name**

The name used to bind one name context into the other; for example,

```
binding direction = local into remote
binding name = cbnt.austin.ibm.com
```

Where the host name tree for the host called *cbnt.austin.ibm.com* is to be bound into the remote name context on the remote host.

**description**

An optional 256-character text field for you to store any text

The following table defines the valid values and resulting scopes that can specified for infrastructure boundary-elements (server, home, and container) of a single location object:

**Note:** The use of **SERVERNAME*, **SERVERGROUPNAME*, and **LOCALNAME* may be depricated in a future release, and therefore it is best if you avoid using these keywords if possible.

*Table 15.* **Single Location Infrastructure Boundary-Element Values/Scopes**

| Server | Container | Home | Resulting Infrastructure Scope |
|---|---|---|---|
| name, *SERVERNAME, *SERVERGROUPNAME | name | name | A specific home in a specific container in a specific server or server group |

*Table 15.* **Single Location Infrastructure Boundary-Element Values/Scopes** *(continued)*

| Server | Container | Home | Resulting Infrastructure Scope |
|---|---|---|---|
| name, *SERVERNAME, *SERVERGROUPNAME | name | *ANY | Any home in a specific container in a specific server or server group |
| name, *SERVERNAME, *SERVERGROUPNAME | *ANY | *ANY | Any home in any container in a specific server or server group |
| *ANY | *ANY | *ANY | Any home in any container in any server or server group |

The following table defines the valid values and resulting scopes that can specified for the topology boundary-elements (cell, work group, and host) of a single location object:

*Table 16.* **Single Location Topology Boundary-Element Values/Scopes**

| Cell | Work Group | Host | Resulting Topology Scope (As a Name Path) |
|---|---|---|---|
| *LOCAL | name, *LOCALNAME | name, *LOCALNAME | /.:/workgroups/<work group-name>/hosts/<host-name> |
| *LOCAL | name, *LOCALNAME | *IGNORE | /.:/workgroups/<work group-name> |
| *LOCAL | *IGNORE | name, *LOCALNAME | /.:/hosts/<host-name> |
| *LOCAL | *IGNORE | *IGNORE | /.: |
| *LOCAL | *LOCAL | name, *LOCALNAME | /work group/hosts/<host-name> |
| *LOCAL | *LOCAL | *IGNORE | /work group |
| *LOCAL | *LOCAL | *LOCAL | /host |

For more information about the valid values and resulting scopes that can specified for the boundary-elements of a single location object, see Component Broker Location Scopes.

c. To apply the changes, and close the Object Editor window, click the **OK** button.

4. Create a Factory Finder model for the factory finder that is to use the single location.

   To do this, on the pop-up menu of the Configuration, click **New - Factory Finder**. If you already have an appopriate Factory Finder model, omit this step.

5. Configure the Factory Finder onto the Single Location. To do this, complete the following steps:

   a. On the pop-up menu of the Factory Finder, click **Drag**

   b. On the pop-up menu of one of the Single Location, click **Configure Factory Finder**

6. Configure the new objects created onto the application that is to use them. (Do this for both the new Single Location and Factory Finder.) To do this, complete the following steps:

   a. On the pop-up menu of the Application model, click **Drag**

b. On the pop-up menu of each of the new objects, click **Configure Application** or **Configure Application (for host)**

Normally, if the single location or factory finder is provided by the application for use on application servers, you use the **Configure Application** action. If the single location or factory finder is provided by the application for use on host name servers (rather than specific application servers), use the **Configure Application (for host)**.

When the Configuration is next activated, the factory finder will be able to use the location defined by the Single Location object.

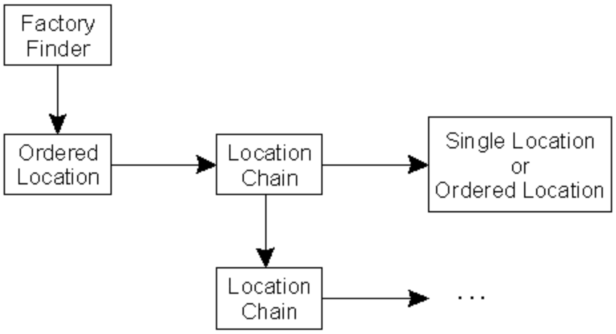
**Related Tasks**

"Configure a Chain of Location Objects"
"Configure a Client Style to use a Specific Factory Finder" on page 412
"Edit Objects" on page 72

## Configure a Chain of Location Objects

Use this procedure to configure a chain (ordered list) of location objects to be used by a factory finder to look for a factory that supports a desired type of managed object, as shown in the figure Use of Ordered Locations (page 234).

**Use of Ordered Locations**



Each location object can be a Single Location or an Ordered Location. A Single Location object defines the location in which to look for a factory.

**To configure a chain of location objects, complete the following steps:**

1. Display the Configuration that contains the applications that are to use the location chain.
2. Create an Ordered Location, to define the start of the location chain. To do this, on the pop-up menu of the Configuration, click **New - Ordered Location**.
3. Create a Location Chain, to represent the first location in the chain. To do this, on the pop-up menu of the Configuration, click **New - Location Chain**.
4. Configure the Location Chain onto the Ordered Location, by completing the following steps:
   a. On the pop-up menu of the Location Chain, click **Drag**
   b. On the pop-up menu of the Ordered Location, click **Configure First in Owned Chain**

5. Define the first location object. For example, to create a Single Location, on the pop-up menu of the Configuration, click **New - Single Location**.

   If the location object is to be an ordered location, create an Ordered Location.

6. Configure the Location Chain onto the Single Location, by completing the following steps:

   a. On the pop-up menu of the Ordered Location, click **Link to Chain**

   **Note:**This assumes that you dragged the Location Chain last, as above. If not, first click **Drag** on the pop-up menu of the Location Chain.

7. Configure the Single Location to define its scope. To do this, complete the following steps:

   a. On the pop-up menu of the Single Location, click **Edit** to display the Object Editor window

   b. Define the scope of the location, by changing appropriate **scope** attributes from the following list:

      **cell scope**
      The cell scope boundary for the single location

      **container scope**
      The container scope boundary for the single location

      **home scope**
      The home scope boundary for the single location

      **host scope**
      The host scope boundary for the single location

      **server scope**
      The server scope boundary for the single location

      **work group scope**
      The work group scope boundary for the single location

   For a table of valid values and resulting scopes that can specified for the topology boundary-elements (cell, work group, and host) of a single location object, see Single Location Topology Boundary-Element Values/Scopes (page 233).

8. Add other location objects to the chain, as required. To do this, repeat steps 3 through 7 for each location object needed in the chain.

9. Relate each Location Chain to the preceding Location Chain in the chain. To do this, for each Location Chain, complete the following steps:

   a. On the pop-up menu of the Location Chain, click **Drag**

   b. On the pop-up menu of the preceding Location Chain, click **Configure Previous in Chain**

10. Create a Factory Finder for the factory finder that is to use the location chain.

    To do this, on the pop-up menu of the Configuration, click **New - Factory Finder**. If you already have an appopriate Factory Finder, omit this step.

11. Configure the Factory Finder onto the chain of location objects. To do this, complete the following steps:

    a. On the pop-up menu of the Factory Finder, click **Drag**

    b. On the pop-up menu of the ordered location object that defines the start of the chain, click **Configure Factory Finder**

12. Configure the new objects created onto the application that is to use them. (Do this for all the new Ordered Locations, Location Chains, Single Locations, and Factory Finders.) To do this, complete the following steps:
    a. On the pop-up menu of the Application, click **Drag**
    b. On the pop-up menu of one of each of the new objects, click **Configure Application**

When the Configuration is next activated, the factory finder will be able to use the chain of location objects.

**Related Tasks**

"Configure a Single Location" on page 231
"Configure a Client Style to use a Specific Factory Finder" on page 412
"Edit Objects" on page 72

## Configure a Factory Finder

Use this procedure to configure a new factory finder.

You need to use this procedure only if you want to define a special scope for factory finding operations. For example, you might create a factory finder whose location identifies one particular server to use when looking for factories.

**Prerequisites:** The factory finder uses a location object to define the scope of factory finding operations. The location must already be configured as a chain of location objects or as a single location, as described in:

- "Configure a Chain of Location Objects" on page 234
- "Configure a Single Location" on page 231

**To configure a factory finder, complete the following steps:**

1. Display the Configuration that contains the applications that are to use the factory finder.

   On the pop-up menu of the Configuration, click **New - Factory Finder**.
2. Configure the Factory Finder onto the location that it is to use. To do this, complete the following steps:
   a. On the pop-up menu of the Factory Finder, click **Drag**
   b. On the pop-up menu of one of the following, click **Configure Factory Finder**:
      - Ordered Location, to use a chain of location objects
      - Single Location, to use the one managed location object
3. Configure the new factory finder onto the application that is to use it. To do this, on the pop-up menu of the Application, click one of the following actions:
   - **Configure Factory Finder**, if the factory finder is to be used on application servers
   - **Configure Factory Finder (for host)**, if the factory finder is to be used on host name servers rather than on specific application servers

When the Configuration is next activated, the factory finder can be used for factory finding operations.

**Related Tasks**

“Configure a Chain of Location Objects” on page 234
“Configure a Single Location” on page 231
“Configure a Client Style to use a Specific Factory Finder” on page 412
“Edit Objects” on page 72

## Configure a new Connection to a Database for use by Applications

Use this procedure to *configure* a connection to a relational database (for example, a DB2/MVS database) for use by one or more applications running on Component Broker application servers.

**Prerequisites:**

You only need to complete this task if an application is to use a connection to a relational database and the characteristics have not already been set within the application family package.

This task involves creating the following system management objects, editing some of their attributes, and creating relationships between them:

- An RDB Connection model, to define the characteristics of the connection to the database
- A Database Alias. If your application uses queriable objects (has a SpecificXXX.ddl file), this defines details of the database.

You also relate these objects to the Application components that use the database connection.

The application should already have been “Add an Application into a Configuration of your Application Environment” on page 228.

If the Database Alias and RDB Connection models already exist, you do not have to create new ones. Likewise, if the relationships already exist, you do not have to create new ones. Database Alias and RDB Connection models and their relationships are normally created automatically from the application’s DDL file when the application is added into the Configuration.

If the RDB Connection model was created automatically when the application was added into the Configuration, it may have a default name. You should consider renaming the Connection model to something unique and appropriate to the use of the connection.

**To configure the database connection, complete the following steps:**

1. Expand the Configuration that contains models for your Application and the Servers that it is to run on.
2. If your Configuration does not already contain an RDB Connection for the database connection, create a new RDB Connection model by completing the following steps:
   a. From the pop-up menu of your Configuration, click **New - RDB Connection**.
   b. In the dialog box displayed, type an appropriate, unique, name for the new RDB Connection. This name is used for system management purposes only. It is an ASCII string and can contain embedded blanks.

c. To create the RDB Connection model, in the dialog box click **OK**.

d. To display the RDB Connection model, expand the RDB Connections folder. You should see an object with the name specified in step ii. If not, repeat the preceding steps.

3. For a new RDB Connection, or if you want to change an existing RDB Connection, set the characteristics of the connection. To do this, from the pop-up menu of the RDB Connection model click **Edit**. This displays the Object Editor for the RDB Connection.
4. In the Object Editor notebook, click the **Main** tab.
5. Set the following attributes to your chosen values:

   **close string**
   The **xa_info** parameter of the **xa_close** request for an XA database. It provides instance-specific information required by the resource manager during termination. For a DB2 database, this attribute is not required and should be left blank.

   **database name**
   The name of the relational database that the connection is to.

   **description**
   An optional 256-character text field for you to store any text

   **flags** Flags controlling the use of the RDB Connection

   **open string**
   The **xa_info** parameter of the **xa_open** request for an XA database. Specify the name of the database that you are using and, if needed, the userid and password that should be used to access that database; for example,

   ```
   mydatabase
   (or)
   mydatabase, myuserid, password
   ```

   Any userid and password specified must be known to the database Security Server and have appropriate authority to access the database:

   a. authority to connect to the database

   b. authority to execute required packages in the database

   c. authority to read and update your application tables

   If you specify a userid and password, the transaction service can open the database using that userid and password. When a database is opened successfully, the remainder of the transaction-related processing has the authority of that userid.

   When you have finished, click **OK** to apply your changes and close the Object Editor window.
6. Relate the RDB Connection model to the Application components that are to use it; the Application, Container, DLL, and Home. To do this, complete the following steps:

   a. On the pop-up menu of the RDB Connection model, click **Drag**.

   b. On the pop-up menu of the Application model, click **Configure RDB Connection**. This creates a *Provided RDB Connections* relationship between the RDB Connection and the Application. You can see the relationship created by expanding the relationship folder within the Application.

c. Expand the *Provided Containers* folder within the Application.

d. On the pop-up menu of the Container that uses the connection to the database, click **Configure RDB Connection**. This creates a *Configured Container* relationship between the RDB Connection and the Container. You can see the relationship created by expanding the relationship folder within the RDB Connection.

   If several containers use the database connection, repeat this step for each Container model.

e. Expand the *Provided DLLs* folder within the Application.

f. On the pop-up menu of the DLL that the application uses to create connections to the database, click **Configure RDB Connection**. This creates a *Implementing DLL* relationship between the RDB Connection and the DLL. You can see the relationship created by expanding the relationship folder within the Application.

g. Expand the *Provided Homes* folder within the Application.

h. On the pop-up menu of the Home that uses the connection to the database, click **Configure RDB Connection**. This creates a *Configured Container* relationship between the RDB Connection and the Container. You can see the relationship created by expanding the relationship folder within the RDB Connection.

   If several homes use the database connection, repeat this step for each Container model.

If several applications use the database connection, repeat this step for each Application model.

7. If your application uses queriable objects (has a SpecificXXX.ddl file), and your Configuration does not already contain a Database Alias for the database, create a new Database Alias model by completing the following steps:

   a. From the pop-up menu of your Configuration, click **New - Database Alias**.

   b. In the dialog box displayed, type an appropriate, unique, name for the new Database Alias. This name is used for system management purposes only. It is an ASCII string and can contain embedded blanks.

   c. To create the Database Alias model, in the dialog box click **OK**.

   d. To display the Database Alias model, expand the Database Aliases folder. You should see an object with the name specified in step ii. If not, repeat the preceding steps.

8. For a new Database Alias, or if you want to change an existing Database Alias, set the characteristics of the database. To do this, from the pop-up menu of the Database Alias model click **Edit**. This displays the Object Editor for the Database Alias.

9. In the Object Editor notebook, click the **Main** tab.

10. Set the following attributes to your chosen values:

    **data base driver name**
    The name of the database driver.

    **data base name**
    The name of the database. This should match the database name specified on RDB Connections to the database.

    **data base type**
    The type of the database; default, DB2/MVS.

When you have finished, click **OK** to apply your changes and close the Object Editor window.

11. Relate the Database Alias model to the *specific* Application (loaded from the application's SpecificXXX.ddl file) that is to use the database, by completing the following steps:
    a. On the pop-up menu of the Database Alias model, click **Drag**.
    b. On the pop-up menu of the Application model, click **Configure Database Alias**. This creates a *Provided Database Alias* relationship between the Database Alias model and the Application model.

       You can see the relationships created by expanding the relationship folder within the Application.
12. If several *specific* applications use the same database, you can configure the same Database Alias model onto each Application model. To do this, repeat step 7 for each other Application model.
13. If an application uses queriable objects in several databases, repeat steps 4 through 7 for each other Database Alias model.

When the Configuration is next activated, the application will be enabled to use the database connections on all servers that the Application models are configured on.

**Related Concepts**

"Database Aliases" on page 28
"Connections to Tier-3 Systems" on page 29

**Related Tasks**

"Chapter 13. Administer Connections to Tier-3 Systems" on page 343
"Change the userid and password used to access a DB2 Database" on page 243
"Change the Database used by Applications"

## Change the Database used by Applications

Use this procedure to change the name, and optionally the userid and password, used by applications to access a database. This procedure is useful when you want to switch an existing database connection from one target database to another.

The transaction service uses the userid and password specified to open the database. When a database is opened successfully, the remainder of the transaction-related processing has the authority of that userid.

**Prerequisites:**

- The application using the database should already have been "Add an Application into a Configuration of your Application Environment" on page 228.
- The package for the application must already exist in both databases. Therefore, before you connect to the new database you should bind the application package to that database using the same bind file as you used to bind the application package to the existing database.
- This procedure changes the database name on an RDB Connection and, if needed for queriable objects, on a Database Alias. All applications that use the same RDB Connection and, if needed, Database Alias will switch from using the existing database to using the new database.
- If you want some applications to continue using the existing database, you should define a separate connection to the new database and configure it onto

the applications that are to use the new database. This is described in “Configure a new Connection to a Database for use by Applications” on page 237.

- If more than one application or container on an application server specifies the same database name (database alias), the first one the server reads is the only one it uses. Therefore, if two applications on the same application server want to use the same database, but each application wants to specify a different userid and password, a different database alias (unique to the database server) must be used. For example:
  - Application1 sets:
    - ...\PolicyXXX\Provided RDB Connections\app1_container

      ```
      database name = App1Policy
      open string   = Policy, App2_userid, App1_password
      ```
    - ...\Specific PolicyXXX\Provided Database Aliases\app1_database (required for queriable objects)

      ```
      database name = App1Policy
      ```
  - Application2 sets:
    - ...\PolicyXXX\Provided RDB Connections\app2_container

      ```
      database name = App2Policy
      open string   = Policy, App2_userid, App2_password
      ```
    - ...\Specific PolicyXXX\Provided Database Aliases\app2_database (required for queriable objects)

      ```
      database name = App2Policy
      ```

**To change the name used to access a database, complete the following steps:**

1. Expand the Configuration that contains models for your Application and the Servers that it is to run on.
2. Expand the Applications folder
3. Expand the Application that defines the application using the existing database.
4. Expand the *Provides RDB Connection* folder
5. From the pop-up menu for the RDB Connection that defines the connection to the existing database, click **Edit**. This displays the Object Editor notebook for the RDB Connection.
6. In the Object Editor notebook, click the **Main** tab.
7. Set the value of the **open string** attribute to the name of the new database and (if needed) the DB2 userid and password for accessing that database. For example,

   ```
   NewClaimDB
   (or)
   NewClaimDB, myuserid, password
   ```

   If you specify a userid and password, the transaction service can open the database using that userid and password. When a database is opened successfully, the remainder of the transaction-related processing has the authority of that userid. If the userid or password is not supplied, the transaction service will default to use the userid or password of the application server process.

   **Notes:**

   - The name specified on the open string is used by the application server to connect to the database.

- The format of the open string depends on the type of database. For the format to use for your database, see the information on open strings provided with the database product.
- Any userid and password specified must be known to the database Security Server and have appropriate authority to access the new database:
  a. authority to connect to the database
  b. authority to execute required packages in the database
  c. authority to read and update your application tables

8. If your application uses queriable objects (has a SpecificXXX.ddl file), you need to specify the name of the new database on the **database name** attribute of the RDB Connection and on its related Database Alias.

   Set the value of the **database name** attribute to the name of the new database, as specified above.

   **Note:** You only need to change the **database name** field your application uses queriable objects. On the RDB Connection, the database name in the **open string** field is used by the application server to talk to the database. Component Broker uses the string in the **database name** field is used as a database alias. If your application uses queriable objects, then the database name fields *must* match. You also need to specify the name of the new database on the related Database Alias, as described in steps below.
9. To apply the changes and close the Object Editor, click the **OK** button.
10. If your application uses queriable objects (has a SpecificXXX.ddl file), you need to specify the name of the new database on the Database Alias.

    To do this, complete the following steps:

    a. Expand the *specific* Application (loaded from the application's SpecificXXX.ddl file).
    b. Expand the *Provides Database Aliases* folder.
    c. From the pop-up menu of the Database Alias model click **Edit**. This displays the Object Editor for the Database Alias.
    d. In the Object Editor notebook, click the **Main** tab.
    e. Set the value of the **database name** attribute to the name of the new database. This should match the database name specified on RDB Connections to the database.

       When you have finished, click **OK** to apply your changes and close the Object Editor window.

When the Configuration is next activated, the application will switch from using the existing database to using the new database on all servers that the Application is configured on.

### Related Concepts

"Database Aliases" on page 28
"Connections to Tier-3 Systems" on page 29

### Related Tasks

"Chapter 13. Administer Connections to Tier-3 Systems" on page 343
"Configure a new Connection to a Database for use by Applications" on page 237
"Change the userid and password used to access a DB2 Database" on page 243

## Change the userid and password used to access a DB2 Database

Use this procedure to change the userid and password used to access a DB2 database.

Normally, when you install CBConnector, you establish a common secure access to your DB2 databases. Use of this procedure is necessary only if you want to use special authority to access a specific database, based on a special userid and password.

The transaction service uses the userid and password specified to open the database. When a database is opened successfully, the remainder of the transaction-related processing has the authority of that userid.

**To change the userid used to access a DB2 database, complete the following steps:**

1. Start the System Manager User Interface and set the User Level to Expert. (Click **View - User Level - Expert** .)
2. Display the RDB Connection that defines the attributes of the database.
3. From the pop-up menu for the RDB Connection, click **Edit**. This displays the Object Editor notebook for the model.
4. In the Object Editor notebook, click the **Main** tab.
5. Set the value of the **open string** attribute to the name of the database and (if needed) the DB2 userid and password for accessing that database. For example,

   ```
   ClaimDB
   (or)
   ClainDB, myuserid, password
   ```

   Any userid and password specified must be known to the database Security Server and have appropriate authority to access the database:

   a. authority to connect to the database
   b. authority to execute required packages in the database
   c. authority to read and update your application tables

   If you specify a userid and password, the transaction service can open the database using that userid and password. When a database is opened successfully, the remainder of the transaction-related processing has the authority of that userid.
6. To apply the changes and close the Object Editor, click the **OK** button.

These changes will be applied to the DB2 database when you next activate the Configuration that contains the RDB Connection.

**Related Concepts**

“The Model World” on page 39

**Related Tasks**

“Change the Database used by Applications” on page 240
“Configure a new Connection to a Database for use by Applications” on page 237

## Start an Application

Applications are normally started automatically when the “Activate a Configuration” on page 256. After their Configuration has been activated, individual applications can be stopped and restarted as needed. Normally, the applications would be restarted by activating the Configuration again. Otherwise, the procedure described in this topic can be used.

To restart an application *after it has been started at least once by activating its Configuration*, complete the following steps:

1. Display the Application Image
2. On the pop-up menu of the Application Image, click **Run**

**Related Concepts**

“Applications” on page 27
“The Image World” on page 40

**Related Tasks**

“Stop An Application”
“Chapter 9. Administer Applications” on page 221

## Stop An Application

To stop an application, complete the following steps:

1. Display the Application Image
2. On the pop-up menu of the Application Image, click **Stop**

**Related Concepts**

“Applications” on page 27
“The Image World” on page 40

**Related Tasks**

“Start an Application”
“Chapter 9. Administer Applications” on page 221

## Example: Install an Application

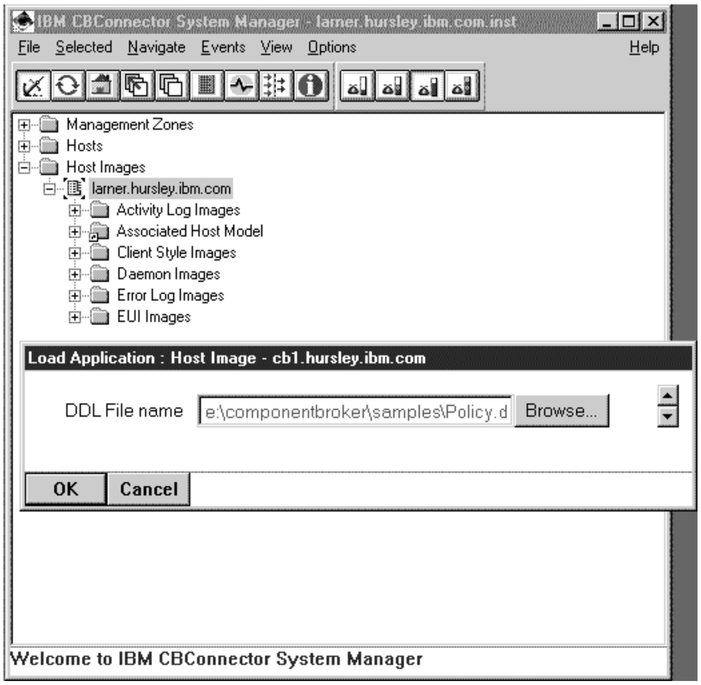
Applications are normally installed by running the installation tool provided with the application. For system management, the application’s installation tool creates the Install objects to represent the application software, based on the DDL files provided with the application.

When those Install objects have been created, an administrator can use them to define the application in the system management model world. The application can then be configured onto servers and made available as part of an activated Configuration.

Another way of creating the Install objects for an application is to use the **Load Application from directory** action on a Host Image. This action performs some of the operations that are normally performed by an application installation tool.

In this example, the **Load Application** action is used to create Install objects from the e:\componentbroker\samples\policy.ddl file. To do this, complete the following steps:

1. On the pop-up menu of the Host Image larner.hursley.ibm.com, click **Load Application**. This displays a dialog box for you to type the name of the directory that contains the application’s DDL file.



DDL File name: e:\componentbroker\samples\Policy.d

Welcome to IBM CBConnector System Manager

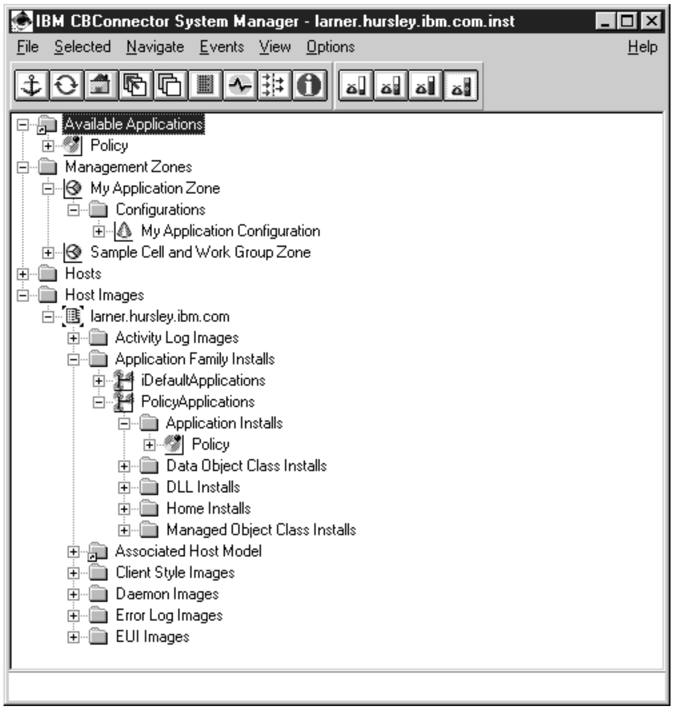
2. In the dialog box, type **e:\componentbroker\samples** (or the name of your own samples directory, if different).
3. To load the application, click the **OK** button. This displays an Action Console window to track the progress and completion of the action.

To see the **Policy** application created by this action, expand the **Available Applications** folder. The Available Applications folder, displayed at the Home of the object network, is convenient for displaying the applications installed on hosts in your enterprise. You can also use the objects in that folder to “Add an Application into a Configuration of your Application Environment” on page 228 within Configurations of your management zones.

To see the corresponding Application Install object, and other Install objects created for the application, complete the following steps. To see the objects created for the application, as shown in Figure 18 (page 4), complete the following steps:

1. Set the user-level to **Expert**; for example, by clicking on the icon of the tool bar. Note that this makes visible the **Application Family Installs** folder for the Host Image larner.hursley.ibm.com.
2. Expand the Application Family Installs folder. You can now see the **PolicyApplications** folder created by the **Load Application** action.
3. Expand the **PolicyApplications** folder. You can now see the folders for the types of objects created by the **Load Application** action.
4. Expand the Application Installs folder, to display the **Policy** application created by the **Load Application** action.

**Figure 18. The Information Controller window**. Showing the sample Policy Application Install created on larner.hursley.ibm.com



To run the application on a server, you need to do the following:

1. “Example: Define an Application” as an Application Model
2. “Example: Configure Applications on Servers” on page 248

Under normal circumstances, there is no need to use the Expert user-level setting, so you can reset the user-level to a lower setting. For example, to set the Basic user-level setting, click the icon of the Tool bar. This displays enough model objects for most applications to be configured and run, and enough image objects for servers and applications to be started and stopped, and statistics gathered.

### Related Tasks

“Edit Objects” on page 72
“Example: Define an Application”
“Example: Operate an Application” on page 250
“Add an Application into a Configuration of your Application Environment” on page 228
“Configure an Application onto a Freestanding Server” on page 186

## Example: Define an Application

This example guides you through using the Information Controller window to define the sample Policy application, which has been added to the **Available Applications** folder. The Policy application can be installed as part of the *Policy Applications* application family or by loading the sample Policy.ddl file, as described in “Example: Install an Application” on page 244.

This example procedure creates an *Application* model to define the Policy application in a Configuration, and configures the application on servers in that configuration.

It assumes that you have the Information Controller window displayed to show the Available Applications folder, as shown in Figure 45.

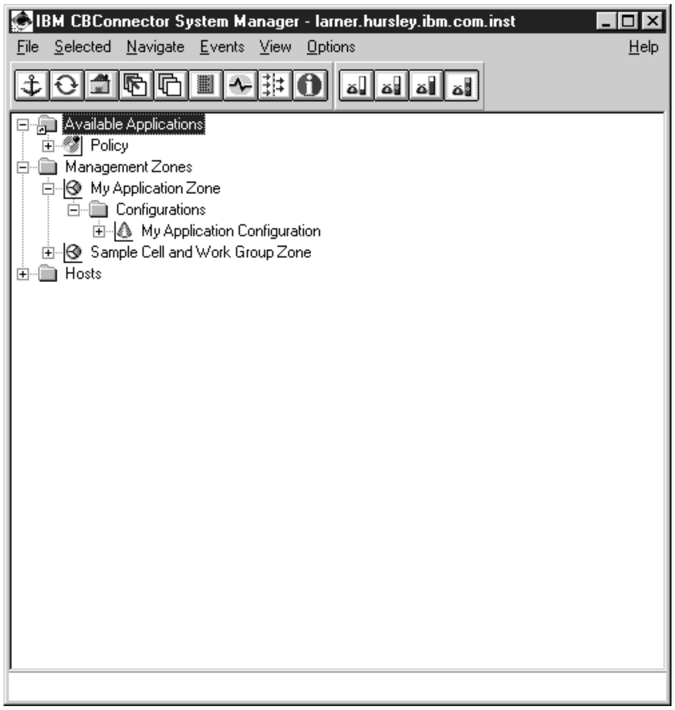

IBM CBConnector System Manager - larner.hursley.ibm.com.inst
File Selected Navigate Events View Options Help
Available Applications
Policy
Management Zones
My Application Zone
Configurations
My Application Configuration
Sample Cell and Work Group Zone
Hosts

*Figure 45. The Information Controller window.*
*Showing the available sample application installed.*

To define the Policy application, complete the following steps:

1. On the pop-up menu of the Policy object, click **Drag.** Note that you can do this on the Policy object in either the Available Applications folder or the Application Installs folder under the **PolicyApplications** Application Family Install.
2. On the pop-up menu of the Configuration **My Application Configuration**, click **Define Application**. This creates an Application model called Policy in the Configuration.

To display the **Policy** Application model created, as shown in Figure 46 on page 248, complete the following steps:

1. Click on the + next to **My Application Configuration**,
2. Click on the + next to the Applications folder

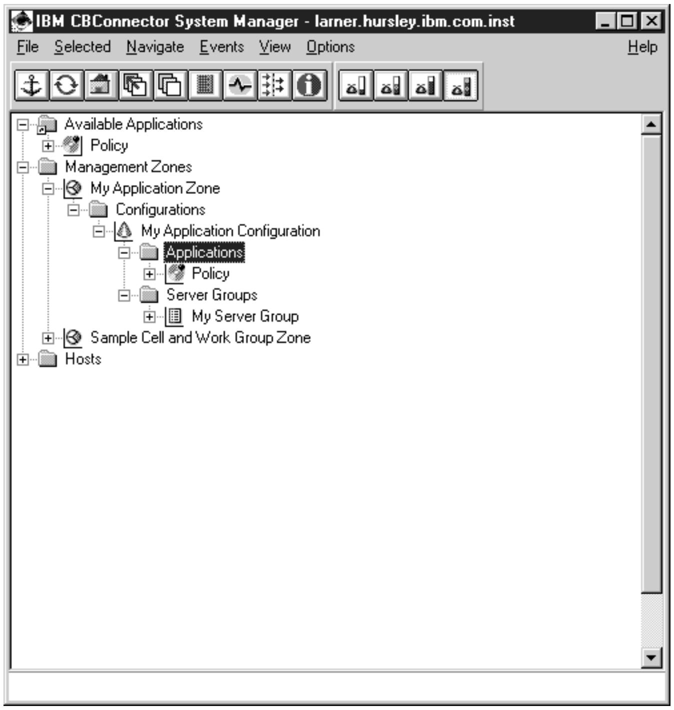

IBM CBConnector System Manager - larner.hursley.ibm.com.inst
File Selected Navigate Events View Options Help
Available Applications
Policy
Management Zones
My Application Zone
Configurations
My Application Configuration
Applications
Policy
Server Groups
My Server Group
Sample Cell and Work Group Zone
Hosts

*Figure 46. The Information Controller window.*
*Showing the Applications folder and the sample Applications that it contains.*

The next step is to configure the application onto the server group (See “Example: Configure Applications on Servers”.), **My Server Group**. This specifies that the application is to be available on the servers that are members of that server group.

### Related Tasks

“Edit Objects” on page 72
“Example: Install an Application” on page 244
“Example: Configure Applications on Servers”
“Example: Operate an Application” on page 250

## Example: Configure Applications on Servers

This example guides you through using the Information Controller window to *configure* the Policy application onto the servers called **My Server 1** and **My Server 2**.

It assumes that you have the Information Controller window displayed to show the Applications model folder, as shown in Figure 47 on page 249. The Applications model folder shown contains the sample Application model created when you “Example: Define an Application” on page 246.

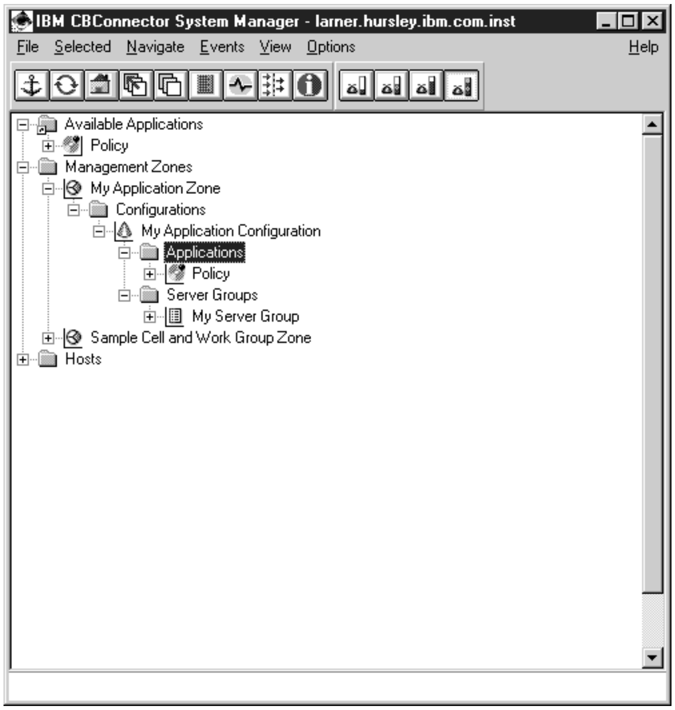

IBM CBConnector System Manager - larner.hursley.ibm.com.inst
File Selected Navigate Events View Options Help
Available Applications
Policy
Management Zones
My Application Zone
Configurations
My Application Configuration
Applications
Policy
Server Groups
My Server Group
Sample Cell and Work Group Zone
Hosts

*Figure 47. The Information Controller window.*
*Showing the Applications folder and the* **Policy** *Application.*

To define that the Policy application is to be available on the servers called *My Server 1* and *My Server 2* you configure the Application model onto the Server Group model, *My Server Group* that the servers are members of. To do this, complete the following steps:

1. On the pop-up menu of the **Policy** Application model, click **Drag**
2. On the pop-up menu of **My Server Group**, click on **Configure Application**

To display the relationship created to represent this, complete the following steps:

1. Expand **My Server Group**
2. Expand the **Configured Applications** folder

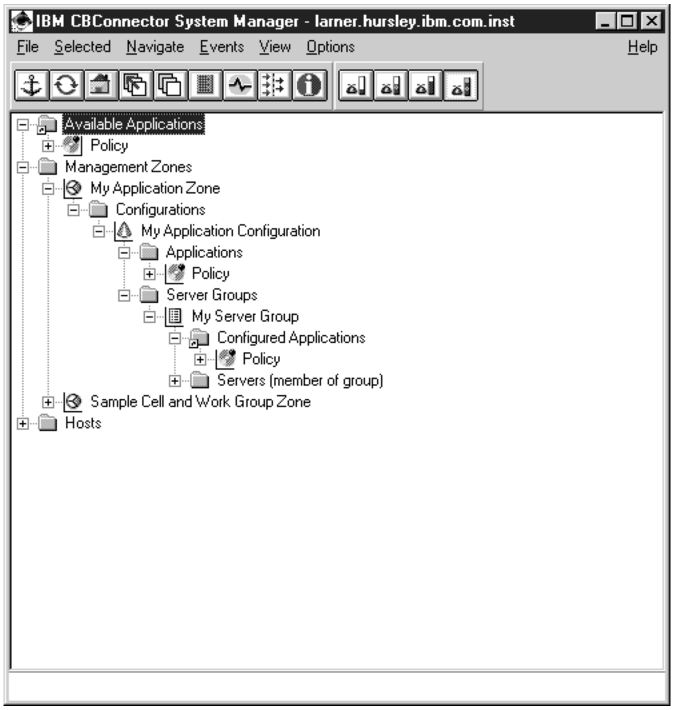


*Figure 48. The Information Controller window.*
*Showing the sample insurance application configured onto the Server Group "pliae server group".*

Now that the application and servers are defined and related, the next step is to verify and activate the configuration..

### Related Tasks

"Edit Objects" on page 72
"Example: Install an Application" on page 244
"Example: Define an Application" on page 246
"Example: Operate an Application"
"Example: Operate a Server" on page 201

## Example: Operate an Application

This example guides you through using the Information Controller window to operate the sample **Policy** application on the running server called **My Server 2**.

It assumes that you have the Information Controller window displayed to show the **Policy** Application Image, as shown in Figure 49 on page 251.

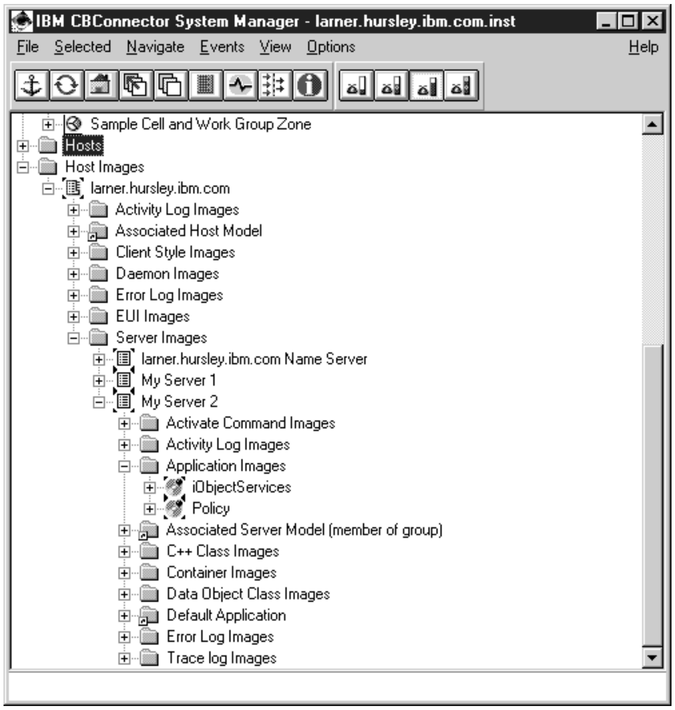


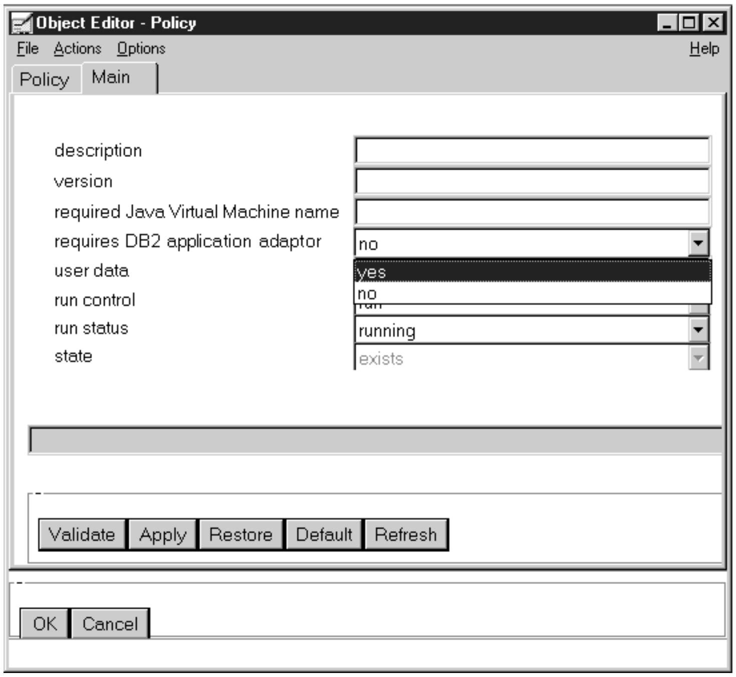
*Figure 49. The Information Controller window.*
*Showing the ″Policy″ Application Image.*

This example changes the **requires DB2 Application Adaptor** attribute to indicate that the Policy application needs to use a DB2 application adaptor.

To change the attribute, you use the Object Editor window as follows:

1. On the pop-up menu of the Application Image, click on **Edit** to display the Object Editor. (See Figure 27 (page Figure 50 on page 252).)
2. The Object Editor displays a notebook that shows a title page listing the path to the **Policy** object and a **Main** tab for the page of attributes.
3. Click the **Main** tab.
4. In the data field of the **requires DB2 Application Adaptor** attribute, click the pull-down icon
5. Click the **yes** value.
6. Click the **Apply** button.

This applies the change, but does not close the Object Editor window.



description
version
required Java Virtual Machine name
requires DB2 application adaptor: no
yes
no
user data
run control: run
run status: running
state: exists

*Figure 50. The Object Editor window.*
*Showing the main notebook page for the ″Policy″ Application Image.*

To stop the Policy application, thereby making it unavailable to users, complete the following steps:

1. Select the **Policy** Application Image. Note that the status bar indicates the state of the application as **Policy: run running**.
2. On the pop-up menu of the **Policy** Application Image, click **Stop Application**. Note that the status bar changes to indicate the state of the application as **Policy: run stopped**.
3. In the Object Editor window, click the **Refresh** button. Note that the data field of the **run status** attribute is now **stopped**.

To restart the Policy application, on the pop-up menu of the **Policy** Application Image, click **Run**.

### Related Tasks

"Edit Objects" on page 72
"Example: Install an Application" on page 244
"Example: Define an Application" on page 246
"Example: Configure Applications on Servers" on page 248
"Example: Operate a Server" on page 201

# Chapter 10. Administer Management Zones and Configurations

This topic provides information about administering the Management Zones and Configurations that you use for your Component Broker host network and the application environment that runs on that network.

You configure your host network within one Configuration of a Management Zone that you should reserve only for configuring your host network. You are recommended to use other Management Zones for your applications. You must activate the Configuration of the Management Zone used for your network before any of your application Management Zones.

You create the Management Zone and Configuration for your host network when you first configure that network, as described in “Create a new Management Zone and Configuration for your Host Network” on page 110.

You can configure all your application environment in one Management Zone. However, you can separate the administration of parts of your application environment into several Management Zones. Each application Management Zone can use unique hosts within your host network or can use the same hosts as other application Management Zones.

You create the first Management Zone and Configuration for your application environment when you first configure the initial application environment, as described in “Create a new Management Zone and Configuration for your Application Environment” on page 139.

You can define alternative Configurations of your Management Zones to cater for changes in your application environment or host network.

The tasks described in this topic can be used whenever required, and are not given in any chronological order.

## Tasks for adminstering Management Zones and their Configurations

The following tasks can be used to administer your Management Zones and their Configurations:

- “Create a new Management Zone and Configuration for your Application Environment” on page 139
- “Add a new Configuration to the Management Zone for your Host Network” on page 170
- “Add a new Configuration to a Management Zone for your Application Environment” on page 254
- “Start or Change an Active Configuration” on page 255
- “Verify a Configuration” on page 255
- “Activate a Configuration” on page 256
- “Delete Objects from an Active Configuration” on page 259
- Activate the System Management Samples

**Related Concepts**

"Administration of Enterprises" on page 6
"Hosts" on page 20
"Cells and Work Groups" on page 19
"The System Name Tree" on page 21
"Remote Name Context Connections to other System Name Trees" on page 23
"The Component Broker Name Tree and DCE" on page 23

**Related Tasks**

"Chapter 7. Administer Application Servers" on page 181
"Chapter 8. Administer Clients" on page 205
"Load a new Application" on page 141
"Chapter 10. Administer Management Zones and Configurations" on page 253
"Chapter 11. Administer Security in your Enterprise" on page 261
"Chapter 13. Administer Connections to Tier-3 Systems" on page 343
"Example: Configure a new Multi-Host Network" on page 121

## Add a new Configuration to a Management Zone for your Application Environment

Complete the steps in this topic to create a new Configuration within a Management Zone used for your application environment.

You create the initial Configuration of an application Management Zone when you first "Create a new Management Zone and Configuration for your Application Environment" on page 139.

The Configuration created in this topic can be used to provide an alternative implementation of an application Management Zone to cater for repeated changes in your application environment.

**To add a new Configuration to an application Management Zone, complete the following steps:**

1. Start the System Manager user interface or, if you are already using it, return to the Home view; for example, by selecting the **Home** icon of the Tool bar.
2. Expand the Management Zones folder
3. On the pop-up menu of your application Management Zone, click **New - Configuration**.
4. On the pop-up menu of your Management Zone, click **New - Configuration**.

   This opens a dialog box for you to specify a unique name for the new Configuration. Type the name that you want the new object to be known by. The name can contain:

   - From 1 through 32 ASCII characters; A through Z, a through z, 0 through 9, underscore (_), and period (.)
   - Embedded blanks
5. To create the new Configuration, click the **OK** button. If you specified a valid name, this creates a new Configuration. If the name specified is not valid, you are prompted to enter a new name.
6. Expand the Configurations folder of your Management Zone. You should see an object for the Configuration that you created. If not, repeat the previous steps.

Next “Load a new Application” on page 141.

**Related Concepts**

“Management Zones” on page 17
“Configurations” on page 18
“Application Servers and Server Groups” on page 25
“Clients” on page 207
“Applications” on page 27

**Related Tasks**

“Create a new Management Zone and Configuration for your Application Environment” on page 139
“Add a new Configuration to the Management Zone for your Host Network” on page 170

## Start or Change an Active Configuration

When you have changed a Configuration, you can **verify** it to check that the changes are valid. When you have completed the definition of a Configuration, you have to **activate** the Configuration to apply it to your enterprise. This updates your enterprise to match the defined Configuration; for example, it starts new servers as defined and stops other servers that were removed from the Configuration.

The following topics describe standard tasks for starting and changing the active configuration of your enterprise:

- “Verify a Configuration”
- “Activate a Configuration” on page 256
- “Delete Objects from an Active Configuration” on page 259

**Related Concepts**

“Configurations” on page 18

**Related Tasks**

“Create a new Management Zone and Configuration for your Application Environment” on page 139
“Add a new Configuration to the Management Zone for your Host Network” on page 170
“Add a new Configuration to a Management Zone for your Application Environment” on page 254
“Start or Change an Active Configuration”
“Verify a Configuration”
“Activate a Configuration” on page 256
“Delete Objects from an Active Configuration” on page 259
“Change the Active Configuration of Your Enterprise” on page 431

## Verify a Configuration

Use this procedure to verify that a Configuration is valid, such that it can be activated to start the application servers, clients, and applications defined in the configuration.

You can verify a configuration at any time *before* you activate the configuration. It is good practice to verify a configuration whenever you have made a significant change; for example, when you have created a new Application Model and configured it onto a Server Model.

To verify a configuration, on its pop-up menu click **Verify**.

Besides checking that your definitions are consistent, the **Verify** action causes CBConnector System Management to create and update some relationships automatically to help you complete the definition of the configuration. If CBConnector System Management finds any errors in the configuration, it displays an error message box to warn you. You should correct the errors before activating the configuration.

The **Verify** action records messages about its action and displays the messages in an Action Console window. The messages record the start and end of the action and significant events that it causes.

When you activate a configuration, CBConnector System Management first completes a **Verify** action automatically, as a final check before starting the **Activate** action.

### Related Concepts

“Configurations” on page 18

### Related Tasks

“Add a new Configuration to the Management Zone for your Host Network” on page 170
“Add a new Configuration to a Management Zone for your Application Environment” on page 254
“Activate a Configuration”

## Activate a Configuration

Use this procedure to activate a Configuration. This creates or update the servers, client styles, and applications defined in that Configuration.

**Before activating a Configuration, consider the following points:**

- Before you activate any of your own Configurations, ensure that the Component Broker name server is running and added to DCE correctly. You can start the name server by activating the sample Configuration of the *Sample Cell and Work Group Zone* management zone provided with CBConnector. These procedures are described in the *Quick Beginnings Guide*.
- Before you first activate your own Configuration, you should decide whether each Server Group that it contains is to be always a controlled server group (for workload management) or basic server group, and configure the Server Group accordingly. (After the Configuration has been activated at least once, you should not switch any server group from its original controlled or basic definition.)
- AIX If any application servers in the Configuration need to use shared libraries that are not in one of the directories specified in the LIBPATH, you must make those libraries available before the application server is started. For information about how to do this, see “Make AIX Shared Libraries Available to Application Servers” on page 191.

- The activation of a Configuration takes place in two phases. In the second phase the SM Agent on each host used by the Configuration will start the Name Server, any Server Group Gateway servers, any Server Group Control Point servers and, finally, any application servers configured to run on that host.
- Before you activate a Configuration, it is good practice to "Verify a Configuration" on page 255. However, when you select the **Activate** action on a Configuration, CBConnector System Management first completes a verifies the Configuration automatically, as a final check before continuing the **Activate** action.
- The **Activate** action used by this procedure completes the following sub-tasks:
  1. Shuts down any servers and clients that are no longer required (not part of the activated configuration). Servers are left stopped with the ″xxx server has been deactivated″ message; you can then remove the server from its host.
  2. Creates and starts any new servers and applications needed by the Configuration.
  3. Copies any application software (for example, DLLs and executable files) needed to a managed host for applications that are configured to run on servers (or client styles) on that host. (You must have already installed the application software onto the System Manager host.)
  4. WIN Adds entries to the Windows NT **Start** menu for client applications that are to be started on managed clients.
  5. For each Client Style model configured on a Host model, produces a client style properties file, which is then stored at the corresponding host. Whether or not the properties file is generated, and where the resulting properties file is placed in the host's file system are controlled by a pair of attributes on the Host model. By default, properties files are automatically generated, and they are placed in **c:\CBroker\data\properties** (on Windows NT) or in **/usr/lpp/CBroker/data/properties** (on AIX).

     If you want to prevent the properties files from being generated, or if you want them placed in a different directory then you can change the corresponding attributes in the main notebook page of the Host model.
- The progress of the **Activate** action is displayed in an Action Console window. When the action completes successfully, you can close that window. If problems occur during the action, messages in the window should help you resolve the problems. For some example sequences of messages, see Example Console Messages for Activating a Configuration (page 257).
- If you need to stop an **Activate** action, you can use the **Stop Activation** action. This leaves the servers, clients, and applications in the states that they were in at the time of the **Stop Activation** action. You should examine the states of the images to ensure that they are appropriate.

**To activate a Configuration, complete the following steps:**

1. Display the Configuration
2. On the pop-up menu of the Configuration, click **Activate**

### Example Console Messages for Activating a Configuration

These example sequences of console messages were output for the following activations of a Configuration that contains 3 application servers called **App Server 1**, **App Server 2**, and **App Server 3**:

1. Messages for the first activation of the Configuration (page 258)
2. Messages for a reactivation (1) of the Configuration (page 258)

3. Messages for a reactivation (2) of the Configuration (page 258)

### Messages for the first activation of the Configuration

```
11/13/98 14:24:29 Verification starting for configuration 'Application Development'
11/13/98 14:24:29 Verification completed – configuration 'Application Development' valid
11/13/98 14:25:05 Activation (host birdland.ibm.com) starting
11/13/98 14:25:15 Activation phase 1 complete for configuration 'Application Development'.
                                                                   1 phase 2 activation initiated.
11/13/98 14:33:46 Starting 'Group 1@birdland.ibm.com Sggw Server'
11/13/98 14:34:49 Server 'Group 1@birdland.ibm.com Sggw Server' startup proceeding, 1 minutes
                                                               (currently at phase 5, step 3 of 3)
11/13/98 14:35:00 Start of Group 1@birdland.ibm.com Sggw Server was successful.
11/13/98 14:35:04 Starting 'Group 1@birdland.ibm.com Sgcp Server'
11/13/98 14:36:05 Server 'Group 1@birdland.ibm.com Sgcp Server' startup proceeding, 1 minutes
                                                               (currently at phase 5, step 2 of 4)
11/13/98 14:36:10 Start of Group 1@birdland.ibm.com Sgcp Server was successful.
11/13/98 14:36:11 Starting 'App Server 1'
11/13/98 14:36:11 Starting 'App Server 2'
11/13/98 14:36:11 Starting 'App Server 3'
11/13/98 14:36:54 Server 'App Server 1' startup proceeding, 2 minutes (currently at phase 2, step 2 of 2)
11/13/98 14:36:55 Server 'App Server 2' startup proceeding, 2 minutes (currently at phase 2, step 2 of 2)
11/13/98 14:36:55 Server 'App Server 3' startup proceeding, 2 minutes (currently at phase 2, step 2 of 2)
11/13/98 14:37:58 Server 'App Server 1' startup proceeding, 3 minutes (currently at phase 5, step 1 of 5)
11/13/98 14:37:58 Server 'App Server 2' startup proceeding, 3 minutes (currently at phase 5, step 1 of 5)
11/13/98 14:37:58 Server 'App Server 3' startup proceeding, 3 minutes (currently at phase 5, step 4 of 5)
11/13/98 14:38:29 Start of App Server 2 was successful.
11/13/98 14:38:30 Start of App Server 3 was successful.
11/13/98 14:38:40 Start of App Server 1 was successful.
11/13/98 14:38:40 Activation (host birdland.ibm.com) completed
```

### Messages for a reactivation (1) of the Configuration

To remove the server **App Server 3** from the enterprise, its Server (member of group) model was deleted from the Configuration, which was then activated again.

These messages were output when the Configuration was activated. Note that servers being removed are started then stopped when needed by the System Manager before the activation of the Configuration has completed.

```
11/13/98 14:43:01 Verification starting for configuration 'Application Development'
11/13/98 14:43:02 Verification completed – configuration 'Application Development' valid
11/13/98 14:44:41 Activation (host birdland.ibm.com) starting
11/13/98 14:44:51 Activation phase 1 complete for configuration 'Application Development' .
                                                                   1 phase 2 activation initiated.
11/13/98 14:47:34 Activation (host birdland.ibm.com): server 'App Server 3' has been deactivated.
11/13/98 14:47:54 Activation (host birdland.ibm.com) completed
```

### Messages for a reactivation (2) of the Configuration

The Configuration was activated without changing anything. The sequence of console messages demonstrates that the server **App Server 3** is not started then stopped when it is not needed. The deactivated server message is still valid.

```
11/13/98 14:49:07 Verification starting for configuration 'Application Development'
11/13/98 14:49:07 Verification completed – configuration 'Application Development' valid
11/13/98 14:51:44 Activation (host birdland.ibm.com) starting
11/13/98 14:51:54 Activation phase 1 complete for configuration 'Application Development' .
                                                                   1 phase 2 activation initiated.
11/13/98 14:54:34 Activation (host birdland.ibm.com): server 'App Server 3' has been deactivated.
11/13/98 14:54:55 Activation (host birdland.ibm.com) completed
```

**Related Concepts**

"Configurations" on page 18

**Related Tasks**

"Add a new Configuration to the Management Zone for your Host Network" on page 170
"Add a new Configuration to a Management Zone for your Application Environment" on page 254
"Verify a Configuration" on page 255

## Delete Objects from an Active Configuration

An active configuration contains model objects that define things in the real world (*real objects*). For example, it can contain a Server (freestanding) model that defines a server running on a managed host.

This topic describes generally how you can delete objects from the real world by deleting them from the active configuration. A convenient way to do this repeatedly, is to create alternative Configurations, with and without the objects. Then, to include an object in the active configuration, you activate the Configuration that contains a model for the object. To remove an object from the active configuration, you activate the Configuration that does not have a model for the object. For example, in this way, you can easily add and remove servers and controlled server groups from the real world.

To delete something from the real world, complete the following steps:

1. Prevent the system management object from being used in a Configuration, by either of the following actions:
   - Delete the model object from the Configuration, by using the **Delete** action from that object's pop-up menu. This specifies that the related Image and real object are no longer required for the Configuration. For example, this would permanently delete a Server (member of group) model from a Server Group or delete an Application model from the Configuration. (To enable the model object to be used again, you would have to define the object again, edit its attributes, and recreate it's relationships as required.)
   - Delete the relationship that controls activation of the model object, by using the **Delete** action from the pop-up menu of the relationship's shortcut icon. This specifies that the related Image and real object are not required when the Configuration is activated again, but means that the model object can be used by relationships with other model objects. Also, to enable the model object to be used again (as specified by the deleted relationship), you need only recreate the relationship. For example, to specify that an application is to be removed from one freestanding server, but still used by other servers, delete the Server model's *Configured Applications* relationship with the Application model.
2. "Activate a Configuration" on page 256. When you do this, the System Manager deletes non-server Images for deleted model objects from managed hosts. For example, if you delete an Application model, then when the Configuration is next activated the Application Images are deleted from the hosts on which the application was available.

   As part of the Configuration's **Activate** action, the System Manager shuts down any objects (for example, servers, client styles, and applications that are no longer required (not part of the activated configuration). For example, this is for

Server models and Application models that you have deleted from the Configuration, and for Application Models for which you have deleted a Server model's *Configured Applications* relationship.

**Caution:** If you delete a Server model from a Configuration, you must also **remove** the Server Image, as described in "Remove an Application Server from a Host" on page 196. This is because, when the Configuration is activated, the server is deleted, but information about the server still exists in the DCE CDS namespace. The **remove** action used cleans up the DCE CDS Name Space.

Likewise, if you delete a Server Group model for a controlled server group from a Configuration, you must also **remove** the Server Images for its application servers, and the Server Group Control Control Point Image and Server Group Gateway Image, as described in "Remove a Controlled Server Group from Your Enterprise" on page 341.

If you want to remove application files from a host, you can use the uninstall function of the application or the **Uninstall** action on the Application Family Install object displayed by the System Manager user interface.

### Related Tasks

"Activate a Configuration" on page 256
"Change the Active Configuration of Your Enterprise" on page 431
"Remove an Application Server from a Host" on page 196
"Remove a Controlled Server Group from Your Enterprise" on page 341
"Remove a Name Server from a Host" on page 176

# Chapter 11. Administer Security in your Enterprise

Component Broker provides the mechanisms and technologies to secure your distributed system. To effectively implement security in a Component Broker network, you need to understand the basic security concepts, how to administer user accounts and certificates, how to setup the various security elements in your topology, and how to perform security administration tasks.

For information about the security concepts used by Component Broker, see “Security in a Component Broker Network” on page 263.

For information about some potential security exposures in this release, see “Security Service Risk Assessment” on page 290.

Use the following set of procedures to create and administer security in your enterprise.

- Although the procedures are listed in a nominally chronological order, individual tasks can be completed as required.
- Where a procedure has a prerequisite dependency on another procedure, or leads on to you completing another procedure, the procedure description provides links to those other procedures.
- You may not need to complete all the procedures, because some are alternatives and some are only needed in special cases, such as if you use SSL-enabled Java clients.

**To create and administer security in your enterprise, complete one or more of the following procedures, as needed:**

- “Plan for Signed Production and Test Certificates” on page 293

  Use this procedure to plan for the signed certificates that you will need to get from a Certificate Authority (CA).

- “Create and Install Server Certificates” on page 292

  This is an overview procedure for a set of procedures that you can use to create a unique certificate for an application server and install that certificate into appropriate client keyrings. You only need to complete this procedure if SSL-enabled Java clients will communicate directly with the server.

  If you want to create and install a test certificate, you can use one of the following procedures:

  – “Use the Test Certificate Provided with Component Broker” on page 295

  – “Create your own Self-Signed Test Certificate” on page 296

  – “Get a Test Certificate from a Certificate Authority” on page 303

  If you want to create and install a production certificate, you should complete the following procedure:

  – “Get a Production Certificate from a Certificate Authority” on page 305

- “Create a Certificate Signing Request and Server Keyring File” on page 299

  Use this procedure to create a *Certificate Signing Request (CSR)*, which you will need to send to a certificate authority (CA) to get a signed certificate for a server from the CA. You only need to complete this procedure if you want to get a signed test or production certificate from a CA.

- “Receive a Signed Certificate into a Server Keyring” on page 307

  You only need to complete this procedure if you a CA has sent you a signed test or production certificate that you want to use with Component Broker.

- “Add a Server Certificate to a Client Keyring Class” on page 310

  Use this procedure to add a server certificate to a client keyring. You only need to complete this procedure if you have a new server certificate that you want to add to a client keyring, to enable clients using that keyring to communicate securely with the server.

- “Place Server and Client Keyrings in your Enterprise” on page 311

  Use this procedure to place server and client keyrings onto appropriate hosts in your enterprise. You need to do this after you have created a new keyring, or have added a new certificate to a keyring, to ensure that the keyring is available on the hosts that need it.

- “Create a Unique Keytab File for an Application Server” on page 313

  Use this procedure to create a unique keytab file for an application server. You only need to complete this procedure if you do not want the server to use the default keytab file created automatically by Component Broker for the server host.

- “Protect Server Keytab Files” on page 315

  Use this procedure to protect the keytab file used by an application server from unauthorized tampering. You should do this for all keytab files used by application servers, including the default keytab file created automatically by Component Broker for each server host.

- Administer Accounts for Client and Server Principals

  Use this procedure to create accounts for all client principals that will use a Component Broker application server, to change account passwords, and if needed to delete accounts. Component Broker automatically creates accounts for server principals, but you can change the keytab file for a server, as described in this procedure.

- “Enable Security within a Configuration” on page 321

  Use this procedure to configure the objects in a system management Configuration so that they can make use of security. For example, you use this procedure to set security attributes on Server and Client Style models.

### Related Concepts

“SSL and Certificates” on page 277
“Certificate Authorities” on page 280
“Accounts for Component Broker Administration” on page 287
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

### Related References

“Security Service Risk Assessment” on page 290

## Security in a Component Broker Network

The following topics provide information aimed at helping you understand the security concepts used by Component Broker, and which you use when administering security in Component Broker networks:

- “Security”
- “Use of DCE- and SSL-Based Security in Your Enterprise” on page 271
- “DCE and the User Registry” on page 274
- “SSL and Certificates” on page 277
- “Accounts for Component Broker Administration” on page 287

**Related Concepts**

“Security”
“Securing your Enterprise” on page 36

**Related Tasks**

“Chapter 11. Administer Security in your Enterprise” on page 261

## Security

The following topics describe general security concepts to help establish a common language for discussing security in Component Broker environments:

- “Trust and Authentication”
- “Appendix B. Quality of Protection” on page 453
- “Delegation” on page 265
- “Encryption Standards, Keys, and Passwords” on page 265
- “Secure Sockets Layer (SSL)” on page 269
- “Distributed Computing Environment (DCE)” on page 271

**Related Concepts**

“Securing your Enterprise” on page 36

**Related Tasks**

“Chapter 11. Administer Security in your Enterprise” on page 261

### Trust and Authentication

All security mechanisms are based on *trust*: trust in some service to properly confirm the authenticity of principals, trust in the security algorithms to perform as advertised, even trust in our own judgements when used to evaluate whether the person in front of us is who they claim and appear to be.

The concept of trust is essential to striking the right level of protection for your information system. To achieve this level you need to be able to answer the question ″who can and should I trust?″.

*Authentication* is the process of establishing that you are who you claim to be; that you are authentic. Component Broker can use Distributed Computing Environment (DCE) and Secure Sockets Layer (SSL) to establish authenticity, as follows:

- DCE establishes authenticity by validating your userid and password against predefined information in the DCE user registry. This is known as *knowledge-based authentication*; you are deemed to be authentic by virtue of the knowledge you possess about your userid, and especially your password which is supposed to remain a secret between you and the user registry. Even system administrators are not supposed to ever know your password. If you keep your password secret, and if noone else is able to guess your password, then the system can assume your authenticity.

  DCE uses a 3-party authentication scheme. To establish the authenticity of a principal, a client or server uses the DCE security server. In this way the server does not have to rely on the trustworthiness of the client software. Likewise, when clients need to verify the authenticity of their servers, the client does not have to rely on the trustworthiness of the server software. If the client host or server host is compromised, the other does not need to trust it to reliably represent the authenticity of the principals they define. Both the client and the server only rely on the trustworthiness of the 3rd-party DCE security server, and the Kerberos/V protocol used by DCE ensures that neither can falsely assert their authenticity. Further, because trust is isolated to only the DCE security server, that server can easily be secured physically; for example, by locking it in a closet or in a closely monitored computing center. By extension, this ensures the integrity of the rest of the computing system that relies on that trust.
- SSL establishes authenticity through the use of public-key *certificates*. Certificates represent a *possession-based authentication scheme*; you are deemed authentic by virtue of possessing a certificate that identifies you. On its own this would not be very secure as, presumably, anyone could manufacture a set of data in the required format that identifies you (or anyone else) and use that to masquerade as you. To prevent this, certificates must be digitally signed by a 3rd-party that vouches for your authenticity. This is similar to what happens with passports. If you present your passport to a Custom’s officer, that officer can use the passport to confirm that you are who you claim to be. The officer is able to discern this, in part, because your passport has a picture of you (something the officer can compare to the person presenting the passport) and in part because the passport itself is embossed in an official package by the national passport office (a neutral 3rd-party). The Custom’s officer doesn’t have to consult some central data repository somewhere to validate some secret that only you should know as happens with the Kerberos 3-party authentication scheme.

  However, certificate-based authentication has a problem in that it lacks a ’picture’. In a distributed computing system there are few, if any, clues that identify the principal that you’re dealing with. It is like having a passport without the picture on it. A Custom’s officer would have to look for other clues to make sure the passport belongs to the person presenting it. Or they have to assume that whomever the passport actually belongs to is reliable and either would never lose their passport or allow it to be stolen, or if they did would be sure to report it. In the latter case, the officer could look up in some list of lost or stolen passports and, if the passport was not on the list, assume the passport must still be acceptable. This presumes that compromised passports are reported to anyone that might need to verify the authenticity of the user, or that there is some other clue that relates the passport to its user.

  In a distributed computing system, it is easier to identify who you are talking to than it is to identify who is talking to you. Moreso, in a Component Broker network, it is easier to ensure that server certificates are not compromised than it is to ensure that client-principal certificates are not compromised. That is because there are typically a lot fewer servers than client principals and therefore fewer things to keep track of. For this reason, Component Broker currently supports only certificate-based (SSL-based) authentication of servers.

**Related Concepts**

"Securing your Enterprise" on page 36
"Security" on page 263

## Delegation

Component Broker does not currently support delegation.

*Delegation* is the process of transferring the credentials of a requesting principal between server processes. In other words, if you invoke a method from your client process to a business object in a server process, that method is executed in the server under your credentials as the requesting principal. If the business object invokes a method on another business object on a different server, you probably want that downstream method to also execute under your credentials. We call the stream of method requests a *method-cascade*, and the process of transferring your credentials to the down-stream method *credentials-delegation*, or just delegation for short.

Delegation can take on several forms. The two most common forms are the following:

- *simple-delegation* (also referred to as *impersonation*) where downstream requests are performed strictly under the identity and authority of the requesting principal (the credential that was delegated).
- *Compound-delegation* involves accumulating the credentials of the requesting principal along with the credentials of all intermediate servers.

Although Component Broker does not currently support delegation, it expects method-cascades to be common in business applications. Therefore, because delegation will become important in Component Broker networks, it plans to support delegation in the near future.

**Related Concepts**

"Securing your Enterprise" on page 36
"Security" on page 263

## Encryption Standards, Keys, and Passwords

The foundation of most security systems is the cipher mechanism that can be used to protect certain information; most notably passwords and other key information, and messages that are exchanged between nodes in a distributed system.

This topic provides the following information about cipher mechanisms:

- Cipher Algorithms and Keys (page 265)
- Types of Cipher Algorithms (page 266)
- Public and Private Keys (page 267)
- Passwords as Cipher Keys (page 267)

**Cipher Algorithms and Keys:**

Cipher algorithms are mathematical formulaes that are designed specifically to obscure the value and content of information. Most valuable cipher algorithms use a *key* as part of their formula. The key is used to encrypt the information, and either that key or a complimentary key is needed to decrypt that information back to a useful form.

The strongest cipher algorithms are those that prevent anyone from deciphering the information they protect without either its cipher key or a tremendous amount of computing power. The strongest cipher algorithms may require months if not years of computing time using the most powerful computing systems to break. These cipher algorithms can then be used to protect the most sensitive information. However, they come at some cost.

Many cipher algorithms increase their protection by increasing the size of the keys they use. However, the larger the key, the more computing time is needed to encrypt and decrypt information; even with the legitimate key. So choosing an appropriate cipher algorithm that strikes a balance between your protection needs, and the computational cost of protecting that information is important.

Another consideration is that while a strong cipher algorithm can thwart deciphering valuable information, given enough time and computational power, any cipher can be broken. As a result, if you do retain information in an encrypted form, it is important to periodically change the key under which it is protected. Typically, you should change your keys more often than the least amount of time it would take the average attacker to break the cipher.

Wherever Component Broker uses ciphers to protect information within the Component Broker network, it often changes the keys used to protect that information. For example, the session keys to protect messages flowing between clients and servers, and between servers and other servers, are recreated for every session. Every time any session is started a new set of keys are created.

**Note:** Servers have passwords and certificate keys, kept in server keytab files and keyring files (see Server Keytab Files (page 276) and “Keyrings” on page 284). It is important that you establish procedures for periodically updating these.

**Types of Cipher Algorithms:**

Cipher algorithms can largely be divided into two categories; those that use symmetric keys and those that use asymmetric keys:

- *Symmetric key ciphers* use the same key to both encrypt and decrypt information. Symmetric key ciphers are valuable because it is relatively inexpensive to produce a strong key for these ciphers, the keys tend to be much smaller for the level of protection they afford, and the algorithms are relatively inexpensive to execute.

  Typically, with a symmetric key cipher, you can exchange the key with another trusted participant; usually you produce a unique key for each pair of participants. You can be assured that any message that you exchange, encrypted in a specific key, between the participants can only be deciphered by the other participant that has that key. In this way, the key must be kept secret to each participant; consequently these are also referred to a *private-key* or *secret-key* ciphers. The major drawback to private-key ciphers is in exchanging the private key, as any exchange has to retain the privacy of the key. This usually implies that any key is also encrypted, but in a different key, because the recipient must already have the key that will be needed to decrypt the key-exchange message. This leads to a never ending dependency on another key.

- *Asymmetric key ciphers* use two separate, but related keys; one is used to encrypt the information and the other key is used to decrypt the information. You cannot use the same key to both encrypt and decrypt the information. So you can treat one key as a *public key* and the other as a *private key*; it does not matter which. This enables you to distribute your public key to anyone. However, by retaining your private key, then anything that anyone else encrypts using your public key can only be decrypted by you. Because asymmetric ciphers involve the use of a public key they are often referred to as *public-key* ciphers. For more information about public and private keys, see Public and Private Keys (page 267).

There are a variety of cipher algorithms. Each falls into one of the two categories indicated above, and each has a set of properties that are worth considering. However, due to their ability to hide information, cipher algorithms are treated by the U.S. government (and certain other governments) as *″controlled munitions″*. The U.S. government allows cipher technologies to be licensed for export, but with certain constraints; primarily that cipher technologies be used with relatively weak keys.

Component Broker currently ships only cipher algorithms that can be exported and used in countries outside of the U.S. This includes RSA for certificate authentication, MD5 and SHA for integrity, and CDMF, RC4 and RC2 for confidentiality. Each of these ciphers are described in more detail in the table Cipher Algorthims Provided by Component Broker (page 267).

**Cipher Algorthims Provided by Component Broker**

| Cipher | Key Orientation | Key Lengths | Description |
|---|---|---|---|
| RSA | Asymmetric | 508-512 bits | Named for authors Rivest, Shamir, and Adleman: Used to digitally sign and authenticate certificates |
| MD5 | One-way Symmetric | 128 bits | Message Digest 5:<br>One-way hash, used to compute digest values for integrity checks. |
| SHA | One-way Symmetric | 160 bits | Secure Hash Algorithm:<br>One-way hash, used to compute digest values for integrity checks. Stronger than MD5. |
| CDMF | Symmetric | 40 bits | Commercial Data Masking Facility: CDMF is an exportable replacement to DES for encryption. |
| RC4 | Symmetric | 40 bits | Rivest Cipher-4: Stream cipher |
| RC2 | Symmetric | 40 bits | Rivest Cipher-2: Stream cipher |

### Public and Private Keys

Asymmetric key ciphers use two separate, but related keys; a *public key* and a *private key*. One is used to encrypt the information and the other key is used to decrypt the information.

These keys work in an asymmetric fashion, in that something encrypted in a public key can only be decrypted with the private key, and vice-versa. This has the following effects:

- I can freely give away my public key to anyone that wants it. Anything they encrypt in my public key can only be decrypted in my private key. If I haven't given away my private key, then anyone can send me information that no-one else can see; even if they also have my public key.
- Conversely, if I encrypt something in my private key and send it out, while anyone with my public key can decrypt it, they can all be assured that only I sent the information. This is because only I could have encrypted it in a way that could be decrypted with my public key.

This illustrates the following two things:

- Communication can be protected in a public domain environment such as the Internet, or really any broadly distributed computing system, without relying on sneaker-net key-exchange procedures.
- More importantly, the security of distributed communication relies heavily on principals not losing possession of their private key. The same thing can be said of any security system, however like all possession-based security mechanisms, the only way to revoke ones ability to enter into the system is to change the locks. These two characteristics basically define the primary benefits and draw-backs of public key authentication.

One application of a public-key cipher is to exchange a symmetric key for use with a private-key cipher. For example, if I want to exchange a symmetric key with you, I could generate an asymmetric pair of public and private keys. I could then publicly broadcast my public key to you; it does not matter that anyone else sees the public key. Upon receipt, you could then generate a symmetric private key, encrypt that key in my public key and send the encrypted result back to me. I could then decrypt that result with my private key to reveal the symmetric key you generated, which we could then begin to use to exchange other messages.

Public-key ciphers are valuable, especially for key exchange. However, they tend to be relatively expensive to execute, their keys tend to be relatively large for the protection they provide, and it is generally expensive to produce a suitable public/private key-pair.

**Passwords as Cipher Keys:**

Passwords are sometimes used as cipher keys; or at least, a one-way function on the password is used to produce a cipher key. Passwords are inherently weak keys; they are both short in length and composed of a small set of potential values. Passwords are often less than eight characters each, small enough to be remembered by people, and are composed only of the alphabetic and numeric values available commonly on computer keyboards, or that can be typed easily. More than that, though, is that passwords are usually created by the people that use them, in part for the following reasons:

- So that only the person creating a password knows its value and thus further protecting the password from being inadvertently disclosed to a malicious rogue.

- To assist the person in remembering their password by allowing them to pick something that is memorable.

Unfortunately, this practice can lead to producing passwords that are even more vulnerable than they might be otherwise. If left to their own, people create passwords that are easiest for them remember, without regard for the consequences that this might have on the enterprise information system. That also means the passwords people pick can be easy for a rogue to guess at and, as a result, break into the information system.

You can control this somewhat by requiring that passwords be composed in a way that prevents them from being simple and guessable. In addition, to avoid typical dictionary attacks, you can establish policies that avoid certain common values, and that cannot be used repeatedly. Finally, you can establish policies that require your end-users to change their passwords frequently; often enough to thwart attacks that involve breaking any ciphers that use those passwords.

Another consideration when you evaluate the cost, risk, and benefits is the amount of time you require end-users to spend in changing their passwords. This should be traded off against the value of the information they deal with. This may mean that you have to establish policies that require users of more sensitive information to change their passwords more often.

However, you should consider another effect; that is, that if the obstacles to using the information system become too great, then people often look for ways to get around them. For instance, if end-users are forced to change their passwords too often, or if those passwords are too difficult to remember, then people will write their passwords down; perhaps on the office calendar. This is equivalent to not having any password at all, because any rogue can easily look around an office, find this information, and then use it to attack the system.

Consequently, even if you think you are increasing the protection of your system, providing greater obstacles can backfire and result in actually decreasing the protection of your system. At that point you may have to resort to psychological or economic approaches to securing your systems; such as making such practices (or avoiding them) a condition of employment for end-users, or instilling a culture that understands and respects the value of the information system used by the business.

**Related Concepts**

“Securing your Enterprise” on page 36
“Security” on page 263

## Secure Sockets Layer (SSL)

*Secure Sockets Layer (SSL)* is an authentication protocol introduced as an IETF standard. Component Broker supports SSL-based authentication between Java clients and application servers. SSL is used to form a secure connection, an *encrypted pipe*, between the client and the server over which the client principal’s userid and password can be passed. The userid and password are then used with DCE to form a Credential representing the authenticity of that principal at the server. This is analogous to *basic-auth* style authentication in common use on the Internet.

The SSL support provided by Component Broker uses the SKit SSL library at the Server, and the SSLight SSL library at the Java client. Because SSLight is written in Java, it can be downloaded from a Web Server along with the rest of the Component Broker client runtime and the client application to support thin clients. You do not have to install anything at the client to perform SSL-based authentication as you do with DCE-based authentication. This is particularly important to Applet-based clients where the primary intent is to be able to download the entire set of client software from a Web Server; including, if needed, the Component Broker client runtime. However, for Java applet clients the SSLight libraries must be available from the Web Server, or for Java Application clients the SSLight libraries must be available at the client, along with the rest of the Component Broker client runtime.

Both the SKit and SSLight libraries are shipped with, and installed automatically by, Component Broker.

Component Broker enables the use of SSL, but only in the following specific fashion:

- SSL can only be used from Java-based clients.
- SSL is used to authenticate the application server and to protect the message traffic between the client and the server.
- Due to the administrative issues pertaining to public-key certificates, SSL is not currently enabled to authenticate client principals. Instead, the client principal will be prompted for a userid and password; these are really DCE identities, and are passed to DCE through the application server.

So, SSL is used to encrypt the transfer of the DCE userid and password between the client and the server, and authentication is actually performed at the server by DCE. This is referred to as *SSL Type-I authentication*. In the future, Component Broker intends to introduce SSL-based authentication that operates on client principal certificates; referred to as *SSL Type-II authentication*.

To enable SSL-based authentication, you must create a server certificate for each server you want to authenticate. These certificates, along with their corresponding private key must be placed in a *keyring* file at the server. The server uses this keyring file to present itself to any clients that want to authenticate the server.

The client must also have a keyring, but the keyring file (actually a Java class) is used to list the authorities that client trusts. More specifically, the client keyring defines the trust-basis accepted by that client. The trust-basis is discussed further in “Models of Trust Validation” on page 282, and can get quite complicated, but for now presume that the client’s keyring either contains the certificates (without the corresponding private key) of the servers they trust, or the certificates of the Certificate Authorities they trust. The keyring at the client must be paired up with the keyrings at the servers that client will use. *This is an area that you should be very careful about.*

For more information about Component Broker use of SSL, see “SSL and Certificates” on page 277.

**Related Concepts**

“Securing your Enterprise” on page 36
“Security” on page 263
“SSL and Certificates” on page 277
“Distributed Computing Environment (DCE)” on page 271

## Distributed Computing Environment (DCE)

*Distributed Computing Environment (DCE)* serves as the security backbone for Component Broker. Component Broker users are registered as DCE users, in the DCE user registry. User authentication is performed using DCE security services. Other than where SSL is used between Java clients and application servers, DCE is used everywhere else to create secure associations between clients and servers, and between servers and other servers.

DCE security is composed of the following two parts:

**The DCE security server**
manages the DCE user registry, and performs the authentication of principals. It acts as a trusted 3rd-party in the authentication process.

**The DCE client**
is used to communicate securely with the DCE security server to authenticate principals and to assert the authenticity of clients to any application servers they use, and from the servers' perspective, to certify the authenticity of any client that use them. In addition, the DCE security client is used to protect the message traffic that occurs between a client and an application server (or between application servers). The DCE security client is accessed through the GSSAPI, a standard interface for general security services.

DCE is required in any Component Broker installation, even if SSL-based security is used with Java clients. You only have to install one DCE security server in your Component Broker network. You can install more than one and set them up to replicate all or parts of either the DCE user registry or the DCE Cell Directory Service (CDS) for use with the Naming service. However, only one security server is needed and setting up replicas, while improving availability, has additional administrative and performance implications.

You have to install the DCE client at every client and server host in your Component Broker network, except you do not have to install the DCE client at Java clients that use SSL-based authentication. If you do install the DCE client at your Java client you have the choice of either using DCE or SSL-based authentication. You control which is used through configuration attributes in the Client Style used by that Java client. (See Configured Clients for more information on setting configuration attributes that will be used at the Java client.) If you enable both DCE and SSL-based authentication, then DCE authentication is performed as that is slightly more secure in a Component Broker network.

**Related Concepts**

"Securing your Enterprise" on page 36
"Security" on page 263
"Secure Sockets Layer (SSL)" on page 269

# Use of DCE- and SSL-Based Security in Your Enterprise

The security mechanisms that you can use depend on what your enterprise contains and how it is structured. This also implies different trade-offs. The following topics describe the security mechanisms that you can employ in different parts of your enterprise, and the implications each of those have:

- Security for Java Application Clients (page 272)

- Security for Java Applet Clients (page 272)
- Security for C++ Clients (page 273)
- Security for Application Servers (page 273)
- Security for Web Servers (page 273)
- Security for Web Browsers (page 274)

**Security for Java Application Clients:**

For Java applications, either DCE- or SSL-based security can be used between the Java client and any application server. As with any use of DCE, if you use DCE-based security between the Java application client and an application server, you must install the DCE client at the Java client. DCE is then used to authenticate the client principal and to protect any messages communicated between the client and server.

If you choose to use SSL-based security between the Java application client and an application server, you do not have to install the DCE client library on the client host. However, you must supply a client keyring and configure that in the Client Style properties file you use at that client. The keyring can be installed at the client host. In addition, you have to create and configure a server keyring at any application server that the client communicates with. Each server keyring should be unique to each server. The client keyring must either contain the public certificates for each server it uses or contain the certificate of whatever certificate authorities were used to sign the server certificates; actually the certificate of any signer in the "Certificate Chains" on page 281. This is also referred to as the *trust-basis* for the client.

**Security for Java Applet Clients:**

You cannot use DCE-based security between Java applet clients and application servers. This is because the DCE client library must be installed at the client where you use it and Java applets cannot use such local DCE client libraries (or any other local system resources).

If you use SSL-based security at the Java applet client, you need to define a client keyring for use by each client. Likewise, as with Java application clients, you must define a server keyring for each of the servers that the client will use, and the client keyring must contain the trust-basis of those servers.

However, unlike Java applications, the client keyring for Java applet clients is installed at the Web Server, and then referenced in the client style properties file, which is also installed at the Web Server. In this way, the same client keyring can be shared by many different clients by merely specifying the same client style profile for each of those clients, or by specifying the same keyring file in each of the profiles used by those clients if more than one profile is used.

Also, because it is likely that a Java applet client will use a Web Server, and given the security constraints imposed by the JVM, you are likely to install a small number of application servers on your Web Server host. (For more information, see Security for Web Servers (page 273).) In this case, the application servers that reside on the Web Server host are the only servers that you need to provide a public key Certificate and, by extension, the only servers on which you have to configure a server keyring for your Java applet clients. (You can overcome the security constraints by digitally signing your applet classes.)

Under these assumptions, the Java applet clients form a secure association with the application servers on the Web Server, the only servers that they can communicate with directly, based on SSL and each server's certificate. The client principal is authenticated through this secure pipe and a DCE-based credential is formed for those principals in the application servers at the Web Server host. Thereafter, all down-stream requests between those and other application servers are performed under the DCE-based credentials.

Of course, if you also have some Java application clients in addition to these Java applet clients, those application clients are not subject to the same JVM security constraints and therefore may require the configuration of server certificates at other application servers.

**Security for C++ Clients:**

You cannot use SSL-based security from a C++ client. (SSL-based security is only implemented for use by Java clients.) Using DCE-based security at the C++ clients means that you must install the DCE client library at the C++ client host.

**Security for Application Servers:**

DCE-based security is always used between application servers and other application servers, so you must install the DCE client library at each server host. In addition, you must create a keytab file for each server to contain its login information; the data that is needed to authenticate the server and provide it a legitimate identity to clients and other servers. In addition, if you are going to use SSL-based security from any Java clients, you must provide a server keyring (containing a Certificate representing the server) for the servers that each of those Java clients use.

Whether or not you use Java applet, Java application, or C++ clients, and irrespective of whether you use SSL-based or DCE-based security, you must provide all of your principals (both end-users and servers) an account in the DCE user registry.

Component Broker-based business objects normally acquire their persistence from a data system; either on the local host, or more likely on a remote host; for example, an S/390 host. The Component Broker runtime — specifically the application adapter to which the object is configured — is responsible for establishing the authenticity of any requests made on the data store for that object's persistent data.

**Security for Web Servers:**

If you have established Java applet clients in your Component Broker network, then you must also establish a Web Server. In addition, you need to install your Java applet classes and the Component Broker client runtime on the Web Server host. If you use SSL-based authentication between your Java clients and an application server, you also need to copy your client-style properties files and client keyrings to the Web Server host.

Given the JVM security constraints for Java applet clients, using a Web Server and downloadable Java applet clients poses a special problem to distributed middleware solutions such as Component Broker. Unless you are willing to digitally sign your downloadable Java classes, including those used within the middleware itself, and authorize the signer of the classes in the keyring at your client, your clients can only communicate back to the Web Server Host. Consequently, you either have to sign all of your downloadable classes, or you have to place the application servers that you want to communicate with on the Web Server host.

Signing applet classes and administering the authorization of those classes at each Java client can be administratively cumbersome. Consult your Java language or Web Server documentation for information on how to sign Java applets and to configure your Web Browser to accept them. On the other hand, if you place application servers on the Web Server host, then keep in mind that from those application servers you can communicate with any other application server in your network. (The Java applet security constraints do not extend to things you do in the Web Server.) In this way, you may be able to multiplex all of your clients' requests to Component Broker business objects through the application servers on the Web Server Host. Those application servers can forward such requests to the business objects on other application servers elsewhere in the network, as appropriate for your business logic and the distribution of those objects in your enterprise.

**Security for Web Browsers:**

You should consider the following points about the security of web browsers:

- The certificates that are used by a web browser to authenticate or authorize the HTTPS (Secure-HTTP) protocol or to authorize signed Java classes are stored in the browser's keyring. The keyrings used by a Web Browser and Component Broker are different and not shared.
- As a result of using a different keyring, if you are using certificates to authenticate end-users to other Web-based applications (principal-certificates to represent the end-user), then the browser probably retains the certificate in a private keyring. The end-user may be required to supply a password to release their certificate from the keyring. Consequently, the end-user could be faced with two (or more logins):
  1. To release their principal certificate from the browser's keyring
  2. To authenticate themselves to Component Broker.
- Web browsers in conjunction with other Web-based applications can introduce a different administration model for setting up keyrings and managing certificates. Moreso, different browsers often impose their own variations on the overall administration model.

### Related Concepts

"Securing your Enterprise" on page 36
"Security" on page 263
"Secure Sockets Layer (SSL)" on page 269

## DCE and the User Registry

Component Broker uses DCE security for all authentication of application servers and for authentication of client principals to servers. This means that all Component Broker users must have accounts created in the DCE user registry. This topic describes the following key concepts relevent to Component Broker's use of DCE security:

- Cells (page 275)
- Administration Policies (page 275)
- DCE Accounts for Component Broker Principals (page 275)

- Server Keytab Files (page 276)

### Cells

DCE uses *cells* to organize administrative domains. In DCE, cells are sometimes also referred to as *realms*.

With respect to security, a DCE cell defines the scope of a common set of security policies. In particular, a cell defines a single logical instance of a *user registry*. All principals registered in the same DCE user registry are part of the same cell, and subject to the same security administration policies established for that user registry.

DCE allows you to have multiple cells and to define trust relationship between those cells. Therefore, a principal can be granted privileges to resources belonging to a foreign cell. **However, Component Broker does not currently expose or exploit multiple-cells.**

### Administration Policies

DCE defines a number of administrative policies that you can create for a cell. These include the following policies:

- The lifespan for an account; the account is disabled after exceeding this established lifespan (time).
- The expiration time of passwords; principals must change their password within this specified expiration period.
- The expiration date of passwords; principals must change their password before this specified expiration date.
- The minimum composition of passwords; specifies the minimum length of passwords, whether they can consist entirely of spaces, and whether characters other than alphanumerics are allowed.
- The maximum renewal time for credentials; the maximum amount of time that a credential can continue to be renewed automatically before the principal must log in again.
- The maximum lifetime for a secure association; the maximum amount of time a secure association can be used continuously between any given client and server pair, or server and server pair, for a given principal.
- Whether or not you want to deploy security server replicas in your distributed system and, if so, how many security server replicas you want to deploy. This policy affects the throughput and availability of your system.

You can control any of these policies through the **DCE Director** or the **dcecp** administration tools available on Windows NT and AIX.

### DCE Accounts for Component Broker Principals

You must create a DCE account for each principal that uses Component Broker. This includes both end-users (client principals) and application servers. Even if your client principals access Component Broker through Java clients and you have enabled SSL for use at those clients, those principals still need a DCE account to use business objects in the Component Broker network.

You can use the **DCE Director** or **dcecp** administrative tools to create these accounts. In addition, you can use the same tools to reset principal passwords, or to reactivate locked accounts in the case that either the principal forgets their password, or someone attempts to break into a principal's account and exceeds the consecutive failed log-in attempts limit.

The process for creating a DCE account for your principals is described in more detail in "Chapter 11. Administer Security in your Enterprise" on page 261.

### Server Keytab Files

A *keytab file* can be used to store the userid and password for a particular principal. Keytab files are normally used to authenticate server principals, but only if this option has been enabled in the system management Configuration, and the keytab file has been created and can be accessed by the server process. They can also be used to authenticate client principals (end-users), subject to the same conditions and procedures as server processes.

A keytab file is a convenient mechanism for storing login information that only a server should possess in order to authenticate itself to the security system. Because the login information is kept in a well-known place, the server can be authenticated automatically. Also, the login information can be automatically updated, which helps to preserve the integrity of the login information and facilitates server automation. (For more information about keytab file management, see the *DCE Administration Reference*; particularly the subcommands of the **rgy_edit** security service commands.)

Component Broker automatically creates a single default keytab file, called **v5srvtab**, for each host. The server principal information for every server on that host is entered into that keytab file. Thus, if different servers on the host are started under different local operating system identities, then each of these identities will have to be enabled to access that keytab file. This may unacceptable if you have some servers that are started manually by different administrators and you do not want them to all have access to the default keytab file. In this case, you must create a unique keytab file for a server and protect it separately. For more information about how to create a unique keytab file, see "Create a Unique Keytab File for an Application Server" on page 313.

The operating system identity under which a server runs depends on how the server is started, as follows:

- If a server is activated automatically, then the server runs under the same operating system identity as the ORB daemon (somorbd.exe) that starts it. In turn, the ORB daemon assumes one of the following local operating system identities:
  - The operating-system identity of the SM agent that started the ORB daemon. If Component Broker is started automatically when the host is started (as by default), the SM agent assumes the identity of the administrator registered with Component Broker in the NT Registry, Otherwise, the SM agent assumes the identity of the administrator that manually started the SM agent (bgmain.exe and bgsrvctl.exe).
  - The administrator that manually started the ORB daemon
- If a server is started manually by an administrator, then the server runs under the same operating system identity as the administrator that starts it.

Each server on the same host can have its own entry in the keytab file and establish its own distributed system identity. Nonetheless, any server started by the same daemon or administrator will inherit the same operating system identity.

Thus, the local-operating system identity of a server depends on your system management Configuration and the process you use for starting the server. Most often, it assumes the identity of the administrator registered with Component Broker in the NT Registry. However, it could be the identity of another administrator if portions of the Host are started manually. You must resolve which administrator's identity is used ultimately to start the server.

It is important to know the local operating system identity under which a server runs, because it is this identity that you must enable to access the keytab file. Any other identity that you enable to this file will have the ability to see and possibly change the DCE userid and password for the server. Thus, the permissions for the keytab should be as restrictive as possible.

When placed in a secure file system and protected with restrictive access policies, the keytab file itself can be maintained safely. The policies governing access to the keytab file should be set so that only the server's identity (the local identity under which the server runs) can access the keytab file.

For more information about how to protect the keytab file, see "Protect Server Keytab Files" on page 315.

**Related Concepts**

"Securing your Enterprise" on page 36
"Security" on page 263
"Secure Sockets Layer (SSL)" on page 269

**Related Tasks**

"Create a Unique Keytab File for an Application Server" on page 313
"Protect Server Keytab Files" on page 315

## SSL and Certificates

The *Secure Sockets Layer (SSL)* protocol has gained in popularity in the Internet industry, primarily because of its use of public-key *certificates* as a means of authenticating principals. These certificates represent a possession-based authentication scheme; you are deemed to be who you claim to be (you are authentic) because you possess an appropriate certificate. The certificate identifies you and, through the encryption techniques used to create it, can be proved to be legitimate.

With SSL and public-key certificates, you trust the *"Certificate Authorities" on page 280* that signed any certificates presented to you. If you do not trust the CA then you do not trust the certificate, and by extension you do not trust the principal it represents. You only need to know about the relatively small number of CAs that you trust. As such, you can avoid building a large, central data base of registered users (a user registry), which is essential in an environment that may consist of millions of end-users, such as the Internet.

**Note:** An important thing to understand is that, because SSL-based authentication is based on possession, anyone who can copy your certificate (actually the private-key that protects your certificate) is able to masquerade as you. For this reason, it is very difficult to use SSL-based authentication to perform delegation; that is, to perform down-stream method requests under your identity and authority. Further, since most enterprise information systems need to control access to their resources on an individual, group, or role basis you often have to create some amount of central (or at least centrally managed) user database information in the form of Access Control Lists (ACLs). Thus the value-proposition that makes SSL so attractive to the Internet may not be so attractive to enterprise computing-based environments.

Signing certificates relies on the use of asymmetric ciphers based on public-key encryption. For more information about public-key encryption, see Public and Private Keys (page 267).

A certificate is your key into a resource. Certificates are signed by an issuing CA and validated either on an individual basis, or on some group basis. The larger the grouping, the more certificates are impacted if the certificate becomes compromised.

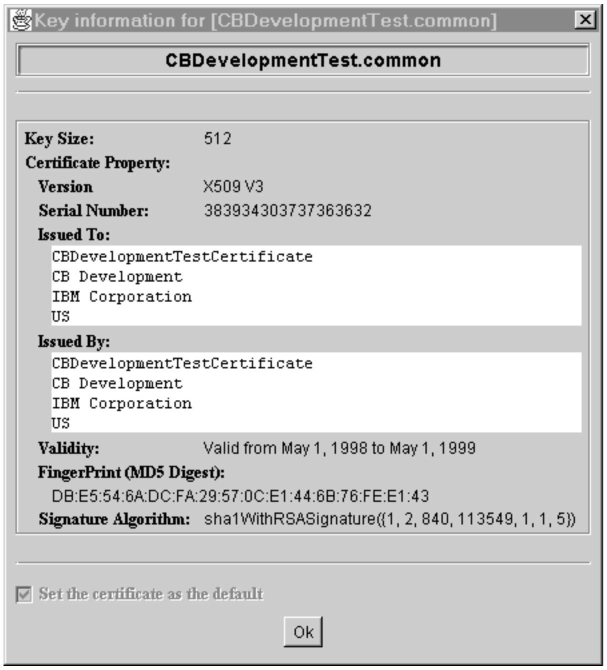
### Structure of a Certificate

A certificate is composed essentially of two major parts; the certificate itself (the public part), and its corresponding private key. As with public-key encryption, you can freely give out the certificate (the public part), if you keep secure the private-key part.

The public portion of the certificate is also composed of two parts; information that identifies you (your name and address), and a *certificate chain*. The certificate chain is the certificate that identifies the authority that issued (signed) your certificate, and the certificate of the authority that signed their certificate (authorized them to be a Certificate Authority), and so on. The certificate chain ends with one or more self-signed certificates; each an authority that authorized itself to be a Certificate Authority. These are known as the *root authorities*.

For more information about certificate chains, see “Certificate Chains” on page 281.

The structure of an X.509v3 certificate is shown in Figure 51 on page 279.



Key information for [CBDevelopmentTest.common]

CBDevelopmentTest.common

Key Size: 512

Certificate Property:

Version X509 V3

Serial Number: 383934303737363632

Issued To:

CBDevelopmentTestCertificate
CB Development
IBM Corporation
US

Issued By:

CBDevelopmentTestCertificate
CB Development
IBM Corporation
US

Validity: Valid from May 1, 1998 to May 1, 1999

FingerPrint (MD5 Digest):

DB:E5:54:6A:DC:FA:29:57:0C:E1:44:6B:76:FE:E1:43

Signature Algorithm: sha1WithRSASignature({1, 2, 840, 113549, 1, 1, 5})

☑ Set the certificate as the default

*Figure 51. The test certificate provided with Component Broker*

### Server Certificates

Component Broker supports the authentication of servers from SSL-enabled Java clients, based on *server certificates*, also referred to as *Site Certificates*. A server certificate uniquely identifies the server.

Even when using public-key certificates to authenticate servers within the SSL-based authentication model, those servers also have DCE-based security credentials. Component Broker application servers are primarily defined as principals in the DCE user registry. Component Broker automatically creates a DCE account for the server when you activate a system management Configuration containing a new server. Creating a certificate for a server is secondary and should only be done if SSL-enabled Java clients will communicate directly with the server. In this case, Component Broker assumes that you have created and installed a unique certificate for each server. There are many choices to make about the procedures you use to generate and maintain server certificates. “Create and Install Server Certificates” on page 292 describes one way to create and administer, but there are many ways for you to tailor these procedures to match the specific needs of your enterprise and its administration policies.

### The Component Broker Test Certificate

Component Broker provides a test certificate that you can use during development or testing so that you can avoid any delays in setting up security for your application servers.

**Note:** *It is very important that you understand that this is an insecure certificate; it is self-signed with a relatively weak key and does not uniquely distinguish the servers where it is used. Therefore, this test certificate should not be used in a production environment where security integrity is required***.**

**Related Concepts**

“Certificate Authorities”
“Certificate Chains” on page 281
“Models of Trust Validation” on page 282
Public and Private Keys (page 267)
“Securing your Enterprise” on page 36
“Security” on page 263
“Secure Sockets Layer (SSL)” on page 269

**Related Tasks**

“Create and Install Server Certificates” on page 292
“Create a Unique Keytab File for an Application Server” on page 313
“Use the Test Certificate Provided with Component Broker” on page 295

## Certificate Authorities

A *certificate authority (CA)* is someone that has been assigned the responsibility for signing your certificates. The process of signing the certificate is an act that warrants the authenticity of the principal represented by that certificate. In other words, the certificate authority has done whatever it takes to ensure that the requesting principal is who they claim to be, and sealed that authenticity by digitally signing the certificate. The CA signs the principal’s certificate with their own (the CA’s) certificate.

Anyone can be a certificate authority. Most often you either trust a particular administrative group within your enterprise to be the CA, or in some cases you may prefer to delegate this responsibility to any one of a number of commercial CAs. A common example of a commercial CA is VeriSign, Inc. (VeriSign can be contacted at http://www.verisign.com).

In the normal process, if you want to obtain a digitally-signed certificate that represents who you are, you begin by producing a *Certificate Signing Request (CSR)* using your local software. You then submit this CSR to the CA, along with any accompanying information that is needed to authenticate you. The CA does what they have to do to verify your authenticity and, if this is successful, signs your certificate and returns it to you. Often, this process can be completed electronically through either e-mail or via the world-wide-web.

However, each CA has their own process. What the CA does to verify you depends on a number of conditions. Ultimately, the reputation of the CA is absolutely essential to their business success. If they fail to properly authenticate a requesting principal properly, accidentally handing out a certificate that the principal then uses to misrepresent themselves, then the CA’s reputation is at stake. You are likely to loose faith in that CA and no longer trust the certificates that they sign. Consequently, principals will soon stop using that CA to sign their certificates, because you have stopped recognizing their authority.

As a result, CAs typically go to great lengths to ensure the authenticity of their requesting principals. They may perform background checks, verify credit histories, post office registries, business records, and even criminal histories.

This process is in many ways equivalent to obtaining a passport; the Passport Office requires that you provide some proof of your legitimacy, such as a birth certificate before issuing you a passport. Other countries accept that the passport proves who you are, trusting that the Passport Office has provided this assurance through its own validation checks and sealed that validity in the physical packaging of your passport.

If you establish your own internal CA (for example, within your systems management group), then you can use less complicated procedures to authenticate certificates. For example, you can verify the requesting principal in your employment records, or based on a previously defined planning manifest; perhaps by listing all of the server principals that you will deploy in your enterprise as part of some major application delivery plan.

**Related Concepts**

"Securing your Enterprise" on page 36
"Security" on page 263
"Secure Sockets Layer (SSL)" on page 269

**Related Tasks**

"Create and Install Server Certificates" on page 292

## Certificate Chains

Often a certificate authority (CA) obtains its authority to sign certificates from another CA. This is particularly common for in-house CAs. In other words, you may deem a certificate signed by a particular CA to be legitimate, not because of who the CA is, but by virtue of that CA having been authenticated by another CA. In this case, the certificate of the requesting principal is signed with the certificate of its CA. That CA's certificate is signed with the certificate of the authorizing CA; the CA that authorized the first CA to sign the certificate for the requesting principal. This is referred to as a *certificate chain*. A certificate chain can be long.

Each successive certificate in a certificate chain represents the next higher *certificate group*. You can even create arbitrary, intermediate CA certificates that you use to sign such groups of principal certificates. Certificate groups are an important element in the organization of hierarchical trust relationships.

For example, you can combine a set of servers into a server group. Assuming you assign each server its own certificate (each representing the server principal of each server) you can sign each of those server certificates with the same common group certificate. The certificate of that server group can then be combined with the signing certificates of other server groups, and all signed with another common certificate representing the super-group.

This can go on until eventually there is one or more certificates that are self-signed. These self-signed certificates are referred to as *root-certificates*; they basically represent the root of the certificate hierarchy below them.

When verifying the validity of a certificate, you must decide who in the certificate chain you are going to trust. You could form your trust in individual certificates, in some intermediate Certificate Authority, or the root authority. We refer to this as the trust-basis for validating certificates. If your trust basis is in individual certificates, then you must retain a list of each individual certificate that you want to recognize. If your trust basis is in the root authority, then you only have to retain the certificate of that authority.

If you ever lose trust in the certificate authority, basically if any certificate issued by that authority is compromised, then you must change the certificate of that authority, and reissue every certificate issued by that authority previously. If your trust was in the root authority, this can be a major exercise. Alternatively, you can reach some balance by establishing your trust-basis in some intermediate authority. (This reduces the impact from losing the trust-basis in that intermediate authority to only the certificates issued by that authority.)

For more information about the effect of certificate chains, and the loss of trust in certificates, see “Models of Trust Validation”.

**Related Concepts**

“Models of Trust Validation”
“Securing your Enterprise” on page 36
“Security” on page 263
“Secure Sockets Layer (SSL)” on page 269

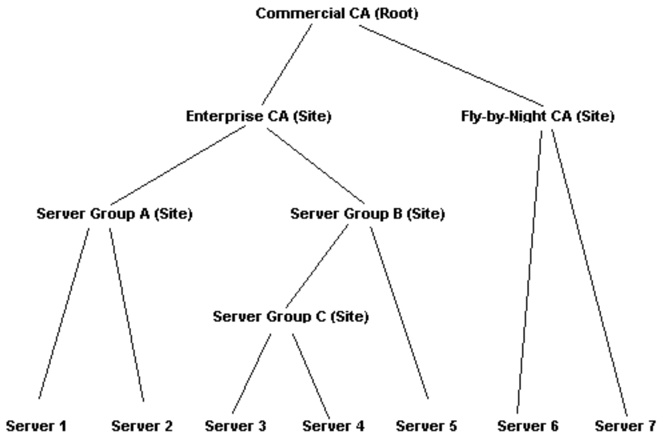
## Models of Trust Validation

At each level in the hierarchy of a “Certificate Chains” on page 281, a certificate derives its authenticity from its next higher-level certificate. A lower-level certificate can only be signed by the owner of the higher-level certificate. Under ideal conditions, you can validate the authenticity of a certificate at any level in the hierarchy that you trust. In general, you do not trust individual client or server certificates. However, you might trust the site or root certificate holder that signed the client or server certificate.

Establishing your trust in the site or root certificate that signed a particular client or server certificate is enough to establish the authenticity of the client’s or server’s certificate. Given the immutable nature of certificates, you only have to determine that the site or root certificate that you trust is in the certificate chain. In this way, you only have to trust a small number of actual entities, and then use that trust to project your trust in the large number of things (clients and servers) that have been signed by them. The entity that you establish trust in is known as the *trust-basis*.

Unfortunately, things aren’t always this simple. Consider the sets of certificate chains depicted in the figure 282. (That figure is used as a basis for discussion throughout this topic.) Let’s say you trust the Commercial CA. Likewise you trust the Enterprise CA, but you don’t trust the Fly-by-Night CA — more particularly you don’t trust the certificates they produce (just because Fly-by-Night was certified by the Commercial CA does not mean they will continue to operate in a trust-worthy fashion). Based on this, you won’t want to accept certificates merely by the fact they were certificated by the Commercial CA. However, you may be able to accept certificates that were signed by the Enterprise CA. Using the Enterprise CA as your trust-basis will result in accepting the validity of all of the servers certified by the Enterprise CA, but not those certified by the Fly-by-Night CA.

**Sample Certificate Chains**



Now consider what might happen if the private key for Server 1 is compromised. Generally, you want to detect the compromised certificate and avoid accepting it if someone tries to authenticate with it. This requires the use of list of certificates that have been revoked. In effect, you will accept the legitimacy of any certificate that has been signed by the Enterprise CA, except those specific certificates that have to been listed in your revocation list.

However, this is a problem for the following reasons:

- Neither SSLight nor SKit currently support revocation lists.
- Even if revocation lists were supported, if a large number of certificates get compromised over the course of their lifetime then the revocation list could get quite large. This is particularly true for the following reasons:
  - Certificates are typically created with an expiration period of a year
  - The revocation list has to be made available to every client (the recipient end of an authentication exchange) in the enterprise
  - The revocation list has to cover every server that a client might access in the lifetime of any one of those servers' certificates.

The problem gets to be even more significant when you consider client certificates in this scenario.

If a server's certificate is compromised, you must renew the certificate of that server's CA, and update your clients to only trust the new CA certificate. This is needed to prevent the server's compromised certificate from being accepted by clients. However, this invalidates the certificates of every server that has been signed by that Enterprise CA certificate. If the Enterprise CA certificate is renewed, the certificate chains for all other servers signed by the same Enterprise CA are broken. Thus, with this model, if any one of those server's certificate is compromised, then every other server's certificate must be renewed with the new Enterprise CA certificate. This can get quite cumbersome in a large distributed system.

One solution is to divide the servers in the enterprise in to logical *server groups*. (This is a normal administrative consideration in Component Broker.) A *Site certificate* can then be issued for each logical server group. Clients can then be set up to accept a server's legitimacy based on their trust in the logical server groups of the enterprise. In this way, if one server's certificate is compromised, then only the certificates of servers in the same server group, and the certificate of the server group itself, need to be renewed. On the one hand, this increases the number of certificates that have to be included in the client's public key ring (indicating the trust-basis for the client), but only by the number of the logical server groups it uses. On the other hand, it reduces the administrative impact if a server's certificate is compromised.

Another solution is to recognize the legitimacy of individual servers; in effect, creating a trust basis with each and every server in the network individually. On the one hand, this provides the greatest administrative freedom; if a server's certificate is compromised, only that server's certificate needs to be renewed. Also, you may be forced to this if you have servers that cannot be logically grouped due to organizational or political constraints. On the other hand, this violates one of the basic tenets of security in large-scale distributed systems; that clients and servers cannot generally be trusted. Also, it requires that a great deal of information has to be replicated; every client has to know the certificates of every server it may communicate with.

However, you may be able to combine both solutions in those cases where a compromise is needed. (Consider the example "Certificate Chains" on page 281 figure.) You could establish trust in Server Group A, Server Group C, and Server 5. In this way, if Server 1's certificate is compromised, then its certificate and Server 2's certificate would have to be renewed, but not the servers in Server Group C or Server 5. Likewise, if Server 5's certificate is compromised, then only its certificate needs to be renewed.

In practice, it is important that the enterprise rigorously maintains its understanding of the relationships between servers and server groups, and its trust in any of those. You would not ever want to get in a situation where Server 3's certificate is compromised and you do not update Server 4's certificate, because you did not realize that their trust basis has been factored to a common server group.

Also, the trust basis needs to be flat. That is, you can have trust in Server Group C and Server 5, but not in Server Group C and Server Group B. This is because, if Server 4's certificate is compromised (and consequently you renew both Server 3's and Server 4's certificates), even after replacing it, the compromised certificate continues to be accepted because of your trust in Server Group B (which continues to show up in the compromised certificate's chain).

**Related Concepts**

"Certificate Chains" on page 281
"Securing your Enterprise" on page 36
"Security" on page 263
"Secure Sockets Layer (SSL)" on page 269

## Keyrings

Component Broker currently uses public-key authentication to authenticate server principals only; not client principals. Client principals are represented by their DCE log-in credentials, not their public key certificates. However, when using SSL-based authentication, clients authenticate the servers with the server's public-key certificate.

Therefore, clients need to have all of the certificates of their trust-basis in any principal they want to allow communication with over SSL. That is, assuming the trust-basis for a client is in individual principals, then the client has to have the public-certificate of each individual principal they want to communicate with. Or, if the trust-basis is in some intermediate or root authority, the client has to have the public certificates for each of those intermediate or root authorities.

Likewise, servers need to have their own certificate and corresponding private keys of any principal they represent.

The certificate information for a client and the certificate and private-key relationship for a server is collected up in *keyrings*.

The keyring for a server should only ever contain one certificate, that of the server's principal, along with its private key. The keyring for a client can contain many certificates corresponding to all of the trust basis it has in the servers it wants to communicate with. Consequently, in general there is a one to many relationship between client keyrings and server keyrings.

However, since several servers can be authenticated by the same CA certificate, then the client keyring can contain far fewer certificates than the number of servers it will communicate with. Using root or site certificates as its trust basis allows the client to factor its trust in one or more CAs over many servers. Further, because many clients can share the same client keyring by using the same "Chapter 8. Administer Clients" on page 205 then there can be a many-to-many relationship between client and server keyrings.

**Note:** You should keep an up-to-date mapping of the relationships between client and server keyrings; even if you do so through manual documentation. If the certificate for any given server is ever compromised, you must invalidate that server's certificate and replace it with a new certificate. If that server is a member of a higher-level trust basis, then you must also replace the certificates for all of the servers and intermediate site certificates within that same trust basis, and then update all of the client keyrings to replace their trust in the new trust basis. *You must be certain to update all of the client keyrings that refer to the compromised trust basis, or the certificates of any sites within the trust basis, lest you be left with a security exposure in your information system.*

### Administration of Keyrings

The most significant administrative task in setting up an SSL-based authentication facility is the administration of keyrings (and the certificates they contain). You must decide who will authorize your server certificates, from the following choices:

- Use a commercial certificate authority
- Use your own certificate authority within your enterprise; possibly whose authority comes from a commercial certificate authority
- Use self-signed certificates.

Based on this decision, you must establish what trust basis that each of your clients is to use. Both of these decisions should be made in the context of achieving a balance between reducing the amount of information you need to make available to your clients against the impact if any of the server certificates are compromised, as follows:

- If you believe there is very low probability of your server certificates every being compromised (or expiring), then you may establish a very high (therefore broad) trust-basis for your clients to reduce the amount of information they have to evaluate when authenticating a server.

- On the other hand, if you have only a few servers, then you may establish a trust-basis in each server and increase the flexibility you have in controlling the authenticity of resulting access each client has in individual servers. In most cases, you can divide your servers into server groups and establish an intermediate trust-basis to achieve an appropriate balance for both limiting information and increasing flexibility.

The client and server keyrings are created using the IKMGUI utility of SKit. The server keyring is generally created as a CMS Key Database while the client keyring is in the format of a Java class file. The IKMGUI tool is discussed further in “Create and Install Server Certificates” on page 292.

### Keyring Passwords

Both IKMGUI and KEYMAN utilities require that you assign your keyrings a password. This is performed in the belief that doing so protects the certificates contained within them. In fact, within a Component Broker network, the keyring passwords are superfluous to the protection of the keyrings.

Client keyrings contain only public certificates, which are not sensitive in a security sense. Nonetheless, the client keyring does define the trust your client has in any servers that is communicates with. Tampering with the client keyring could result in convincing the client that it can trust a server that it otherwise would not. For this reason, the client keyring should be protected. In a Component Broker network keyrings are protected by either the security available to the Web Server, or with a secure file system at your client host.

A server keyring does contain the private key for the server’s certificate and so the keyring should be protected. However, a secure file system at your server host should be used to provide this protection. If the client keyring is protected by its Web Server or the local secure file system, then the keyring password is superfluous.

Likewise, if the server keyring is protected by a secure file system at the server, then again the keyring password is superfluous. For convenience, because you are required to supply a password anyway, the password of **CBroker** is used by convention. If you choose to use a different password, then you should set that password in the corresponding attributes of either the Client Style model or Server model in your system management Configuration. For more information about setting passwords and other security attributes, see “Configure Security for a Server” on page 323 or “Configure Security for a Client Style” on page 326.

### Related Concepts

“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263
“Secure Sockets Layer (SSL)” on page 269

### Related Tasks

“Create and Install Server Certificates” on page 292
“Configure Security for a Server” on page 323
“Configure Security for a Client Style” on page 326

# Accounts for Component Broker Administration

This topic describes the accounts that must established to install Component Broker and its related subsystems.

- "Windows NT Administrator's Account"
- "AIX Administrator's Account"
- "DCE Cell Administrator's Account"
- "DB2 Administrator's Account" on page 288
- "Component Broker Administrator's Account" on page 288
- "Windows NT Workstation Integrated Login" on page 288
- "Windows NT Domain Integrated Login" on page 289

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

## Windows NT Administrator's Account

To install Component Broker on Windows NT you must have administrator authority; that is, the Windows NT userid you use must be a member of the Administrators group. If you will be using Component Broker within a Windows NT domain, you must be a member of the Domain Administrators group.

If the workstation will be used within a Microsoft Windows Networking Work Group, use the Windows NT User Manager to define the userid and ensure that the new userid 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 userid and ensure that the new userid is a member of the Domain Admins group. Then 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 userid (without the domain qualifier) is a member of the local Administrators group. After this verification, log on again, but use the domain-qualified userid.

In general, it is a good idea to install Component Broker, DCE, and DB2 under the same user account. When selecting a user id for this account, consider the stricter "DB2 Administrator's Account" on page 288.

## AIX Administrator's Account

To install Component Broker on AIX you must have root authority.

## DCE Cell Administrator's Account

When you install and configure DCE, you establish a cell administrator's account in the DCE user registry. By default, DCE uses the security name **cell_admin** for this account. Having created this account, and before installing DB2 and Component Broker, if you will be using integrated login, it is a good idea to create another administration account with the same authority as **cell_admin**, and whose security name is consistent with the stricter constraints of the DB2 administrator's id. Alternately, you can specify a different account name during DCE setup, to use in place of **cell_admin**, and with the same constraints imposed by DB2.

For more DCE-account considerations, see the following topics:
- "DB2 Administrator's Account"
- "Windows NT Workstation Integrated Login"
- "Windows NT Domain Integrated Login" on page 289

## DB2 Administrator's Account

If you do not want to use the same account to install DB2 as you use to install DCE and Component Broker, you must create a separate account for DB2. The DB2 administrator's account is subject to following rules:
- It must have Administrator authority, on Windows NT; that is, it must be a member of the Administrator group.
- It must contain fewer than eight 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*.

Other DB2-userid considerations are given in the following topic:
- "Windows NT Domain Integrated Login" on page 289

## Component Broker Administrator's Account

The Component Broker System Manager or a System Management Agent on a host runs as the CBConnector service, under its own account. Component Broker uses this account to create and register the server principals for any servers that it creates when you activate a configuration.

During the installation process, Component Broker will prompt you for the userid and its password. You can either let the installation process create the account, or you can create the account beforehand. If you create the account beforehand, it is important that you give it the advanced user right of ″logon as a service″. (To do this, you can use the User Manager with the **Show Advanced User Rights** option selected to choose this right.)

## Windows NT Workstation Integrated Login

DCE supports integrated login with Windows NT Workstations. Whenever, you login to Windows NT, an attempt is made to log in to DCE at the same time. This function uses the same userid and password for both NT and DCE, and so these must be set the same initially. Thereafter, whenever you change your Windows NT password, an attempt is made to change your DCE password and to ensure these are kept synchronized.

Since the login attempt to DCE and the change password attempt to DCE can fail for a variety of reasons, integrated login will forego these if any problems occur. If you use a DCE-related service (such as Component Broker administration, operations on the Naming service, or certain interactions that require the use of the

Security service) and are denied from using that service, it may be because the integrated login didn't succeed with DCE. You may have to manually log-in to DCE using the **dce_login** command.

You can enable integrated login by using the DCE Setup administration tool, as follows:

1. Display DCE Setup
2. Select the **Configure** menu-bar choice
3. Select the **Modify Configuration - Enable Integrated Login** menu option.

Remember that Windows NT Workstation is a stand-alone, albeit multi-user platform. Thus, if you have two or more Windows NT Workstations, and you intend to move between the two workstations, sometime using one and sometime using the other, the Windows NT user registries are completely separate from each other. If you want to enable integrated login at each of these workstations you will have to ensure the userid and password are the same on both workstations. In addition, when you change your password on one workstation, you will have to change your password to match on the other workstation.

## Windows NT Domain Integrated Login

DCE supports integrated login with Windows NT Domains. Whenever, you login to Windows NT, an attempt is made to log in to DCE at the same time. This function will use the same userid and password for both NT and DCE and so these must be set the same initially. Thereafter, whenever you change your Windows/NT password, an attempt is made to change your DCE password as well to ensure these are kept in sync.

Because a login attempt and change password attempt to DCE can fail for a variety of reasons, integrated login will forego these if any problems occur. If you use a DCE-related service (such as Component Broker administration, operations on the Naming service, or certain interactions that require the use of the Security service) and are denied from using that service, it may be because the integrated login did not succeed with DCE. You may have to manually log in to DCE using the **dce_login** command.

You can enable integrated login by using the DCE Setup administration tool, as follows:

1. Display DCE Setup
2. Select the **Configure** menu-bar choice
3. Select the **Modify Configuration - Enable Integrated Login** menu option.

Windows NT workstations in the same domain share a common Windows NT user registry. In this way, your userid is the same on all workstations in that same domain. Whenever you change your password on one workstation, your password is changed for any other workstation you log in to.

However, you must take care when using DB2 in a Windows NT Domain. DB2 does not recognize domain names; it will truncate the domain prefix from a fully qualified user name. Therefore, you must ensure that your DB2 principal name matches the unqualified user name of your Window NT account in the domain. This can be somewhat problematic if you first install DB2 on a non-domain attached Windows NT Workstation, then later upgrade that workstation to participate in a domain. You must ensure your unqualified user name remains the same after you transition to the domain. For DB2-account related considerations, see the following topic:

- “DB2 Administrator’s Account” on page 288

**Related Concepts**

“Securing your Enterprise” on page 36
“Security” on page 263
“Security in a Component Broker Network” on page 263

**Related Tasks**

“Administer Accounts for Client and Server Principals” on page 316
“Chapter 11. Administer Security in your Enterprise” on page 261

## Security Service Risk Assessment

There are some potential security exposures in the Release 2.0 Component Broker Security Service that have been identified in both DCE and SSL security mechanisms. However, each potential security exposure has a work around or can be avoided if the other option is chosen.

The potential security exposures are described below, grouped by DCE and SSL security mechanisms in the following topics:

- DCE Security (C++ Server, C++ Client, and Java Client) (page 290)
- SSL Security (Java Client) (page 291)

### DCE Security (C++ Server and C++/Java Client)

1. Exposure of the Key Tab files

   The keytab file contains the DCE principal names and passwords of the server processes which are very critical security information to the DCE security. The integrity of the keytab file should be protected by the operating system’s file system so that only the server process can access to the keytab files, as described in “Protect Server Keytab Files” on page 315.

   If the keytab files are not protected by the operating system’s file system, the contents of the keytab files could be corrupted by unauthorized users.

2. Environment Variable Password Exposure

   The Component Broker Security Service allows users to create a credential for a principal by specifying the principal’s userid and password in environment variables.

   However, this approach could introduce a potential security exposure if the userid and password are recorded in a script file in clear text form.

   This warning has been described in the ″Using Environment Variables to Establish Authenticity″ section of the *Advanced Programming Guide*.

3. Operating System Resource Exposure in Server Environments

   The server process assumes the local operating system identity of the Component Broker system processes such as the ORB daemon or the System Management Agent process.

   In an application server, the application code is executed in the address space of the server process with the local operating system identity of the Component Broker system process. Therefore, the application code is eligible to access to the local operating system resource which is authorized to the server process.

There is a potential security (integrity) exposure that the application code might corrupt the system resource in the local operating system.

This warning is also mentioned in the *Advanced Programming Guide*.

4. CICS and IMS userid and password are stored as clear text in CDS

   To support the *get_mapped_security_info* method of the IExtendedSecurity:: Credentials interface implementation, **data system principal** and **data system password** attributes are provided on the Security Service notebook of Server Groups, Servers (freestanding), and Server Images.

   **data system principal** is the attribute that specifies the CICS or IMS userid in clear text. **data system password** is the attribute that specifies the CICS or IMS user passwords, in clear text.

   It is an obvious security exposure to enter passwords in clear text, because there is no protection to prevent unauthorized users from stealing your passwords.

5. Multiple Credential Object Client

   The Component Broker Security server assumes that the server security session id is valid for the entire process life of its counter client process.

   When a client and a server successfully establish a security association, the server security session id is kept in the client process's security Vault Table. The secure server receives the secure session id in the security context to identify the client process.

   However, if the client program changes its credential by invoking a ″request_login″ on a LoginHelper object, it can still continue to communicate with the secure server with the secure server's session id. Note that the server security session id is granted to the client process with the previous credential.

   The application server doesn't continuously verify the DCE credential handle of the client process after the security association between the client and the server is established.

   This exposure can be a problem for a multiple threaded client application which assumes that each thread can have its own credential.

6. DCE Credential Protection

   If the DCE credential that is created by ″dce_login″ is no longer needed, it is a good idea to destroy it with the DCE ″kdestroy″ command so that another application which is running on the same machine cannot use the DCE credential to avoid any security exposure.

**SSL Security (Java Client)**

1. Server Keyring File Exposure

   The Server Keyring files (″server_name.kdb″) that are stored by default in the *CBroker\data\keyrings* directory should be protected with the local operating file system so that unauthorized users cannot remove or corrupt the Server Keyring files.

2. Client Keyring Class Exposure

   The Client Keyring Class file (for example, *CBDevTestClKeyRing.class*) that is stored in the CBroker\data\keyrings directory should be protected with the local operating system so that it can be protected from attacks by unauthorized users. The Client Keyring Class contain the public portion of the target server's certificate.

3. Server Certificate is compromised

   If the Server Certificate is compromised, a new Server Certificate should be generated for the server and stored in the Web server. The new Server

Certificates should be downloaded from the Web Server and the Client Keyring Class should be updated with the new Server Certificates.

This is not an security exposure. However, the client should have this recovery process in place when the Server Certificate is compromised.

4. Server Certificates should be protected in the Web Server

   The Server Certificates are stored in the well known directory in the Web Server. To protect the Server Certificates, it should be protected by the local operating system's file system.

5. Download Server Certificates with security protocol, HTTPS

   The Server Certificates should be downloaded form the Web Server with a secure protocol such as HTTPS so that the Server Certificates cannot be intercepted and corrupted by unauthorized users.

6. Certificate of Certificate Authority is compromised

   If the Certificate of the Certificate Authority is compromised, all Certificates that are issued by the Certificate Authority are compromised. Moreover, all Certificates that issued by the subordinate Certificate Authority of the Certificate Authority are compromised as well.

   This is not a security exposure, but the administration of the Certificate Trust Base should be set up to discover the compromise of the Certificate Authority and recover from this compromised security.

**Related Concepts**

"Security in a Component Broker Network" on page 263
"Securing your Enterprise" on page 36

**Related Tasks**

"Chapter 11. Administer Security in your Enterprise" on page 261

## Create and Install Server Certificates

Use this set of procedures to create a unique certificate for an application server or name server.

You only need to complete this procedure if SSL-enabled Java clients will communicate directly with the server. In this case, Component Broker assumes that you have created and installed a unique certificate for each server.

This topic describes one set of procedures that you can use to create and install server certificates. You can use your own procedures, or tailor the procedures described to match the specific needs of your enterprise and its administration policies. For example, you can get a server certificate from your own Certificate Authority instead of from Verisign.

You can create and install either a *test certificate* for use during development and testing or a *production certificate* for use in a production Component Broker network. Creating test and production certificates follow the same overall procedure, and share some common sub-procedures. (Where used, the common sub-procedures are referred to from within the following procedures.)

**Notes:**

- Depending on how you organize the administration of your certificates, particularly if you involve a commercial Certificate Authority, the time it takes to create and install a certificate for your servers can be significant; perhaps several days. Therefore, you should plan to complete this procedure some days before you need to use the certificate for a server.
- Component Broker automatically creates a Name Server on every host that you configure. If you will be accessing the Name Server from Java clients over the SSL-based authentication facility, you must provide a certificate for this server and include its trust-basis in the client keyrings for any clients that will access it.

If you want to create and install a test certificate, you can use one of the following procedures:

- “Use the Test Certificate Provided with Component Broker” on page 295
- “Create your own Self-Signed Test Certificate” on page 296
- “Get a Test Certificate from a Certificate Authority” on page 303

If you want to create and install a production certificate, you should complete both the following procedures:

1. “Plan for Signed Production and Test Certificates”
2. “Get a Production Certificate from a Certificate Authority” on page 305

**Related Concepts**

“SSL and Certificates” on page 277
“Certificate Authorities” on page 280
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

“Plan for Signed Production and Test Certificates”
“Use the Test Certificate Provided with Component Broker” on page 295
“Create your own Self-Signed Test Certificate” on page 296
“Create a Certificate Signing Request and Server Keyring File” on page 299
“Get a Test Certificate from a Certificate Authority” on page 303
“Get a Production Certificate from a Certificate Authority” on page 305
“Receive a Signed Certificate into a Server Keyring” on page 307
“Add a Server Certificate to a Client Keyring Class” on page 310.
“Place Server and Client Keyrings in your Enterprise” on page 311

## Plan for Signed Production and Test Certificates

Use this procedure to plan for the signed certificates that you will need to get from a Certificate Authority (CA).

In a production Component Broker network, the production certificates are authenticated to verify the principal using the certificate. The principal is actually authenticated by a CA when the CA signs the principal’s certificate.

Because of the diligence that is expected of the CA, as described in “Certificate Authorities” on page 280, the authentication process for principals can take a significant amount of time. Commercial CAs often require up to a week to complete their authentication process. Even on-site CAs can take several minutes, if not hours, or even days, to complete their authentication process.

As a result, when planning to add a new application server or host (name server) to your enterprise you must plan for the certificates that you will need well in advance of actually creating the server. Although primarily a consideration for production certificates, these points apply equally to signed test certificates provided by CAs.

Consider the following points when planning for a new server certificate:

- On the certificate signing request that you send to the CA, you will need to specify the common name for the certificate. This is the primary, universal identity for the Certificate; that is, it should uniquely identify the principal that it represents. Since, in a Component Broker environment, certificates are used to represent server principals, the common convention is to use the form <host_name>/<server_name>. For example, for the server **PolicyServer1** on the host **centralops.xyz.com** you would specify the common name **centralops.xyz.com/PolicyServer1**.

  For some CAs, including the fully-qualified name of your host in the common name is more than just a convention. For example, VeriSign will not sign your certificate unless the domain portion of the host name is owned by your organization. So when planning the common name for a certificate request, check the format that your CA requires.
- On the certificate signing request that you send to the CA, you will need to specify the name and address of your organization. Some certificate authorities, including VeriSign, require that you spell out completely the state or province fields. For example, you need to specify California as opposed to CA. So when planning the address for a certificate request, check the format of address fields that your CA requires.

If you do not get a production certificate for a server (from the CA) before you want to start using the server, you can plan to do either of the following, less secure, alternatives:

- You can use the test certificate provided with Component Broker to perform some early tests. However, given the lack of security implied by that test certificate, you should replace it with a certificate that legitimately represents your server as soon as possible. (For this, you can create your own test certificate for the server.)
- Alternatively, you can configure the server initially without its certificate keyring. This means that your clients cannot access the server securely.

When you receive the server’s keyring (from the CA) you can change the attributes of the server to use its certificate. From then on, the clients can access the server with the security provided by the certificate.

Note that if your server’s certificate is compromised, or even if some other server in its trust-basis is compromised, and you have to produce a replacement certificate, you will experience the same delay again until a new certificate is received.

For more information about getting and installing server certificates, see “Create and Install Server Certificates” on page 292.

**Related Concepts**

“SSL and Certificates” on page 277
“Certificate Authorities” on page 280

“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

“Create and Install Server Certificates” on page 292

## Use the Test Certificate Provided with Component Broker

Use this procedure to install the test certificate provided with Component Broker for use by servers and their SSL-based Java clients.

Component Broker provides a test certificate so that you can avoid any delays in setting up security for your application servers.

**Note:** *It is very important that you understand that this is an insecure certificate; it is self-signed with a relatively weak key and does not uniquely distinguish the servers where it is used. Therefore, this test certificate should not be used in a production environment where security integrity is required***.**

Given the lack of security associated with this test certificate, you should replace this certificate with one that is specific to your server *as soon as possible*.

If you want to use this certificate during the initial setup of your server you must manually copy the certificate into place. Component Broker deliberately avoided automating this process to reduce the potential for this certificate being used inadvertently in production environments.

**To set up the test keyring for use by your server, complete the following steps:**

1. Copy the test certificate called **CbDevTest.kdb** in the Component Broker keyrings directory ( WNT by default, *c:\CBroker\data\keyrings*) to a new file in that same directory.
2. Rename the new file to the the server’s security name, with the same .kdb file extension. (The name should exactly matches the security name you establish for the server in it’s DCE account.)

**To set up the test keyring for use by your client, complete the following steps:**

1. Copy the **CbDevTestClKeyring.class** file to your client host (if using a Java application client) or to your Web Server (if using a Java applet client).
2. Ensure that the client’s **classpath** or **codebase** includes a path to wherever you put this class file.
3. Ensure that the client style properties file that you will use with this client refers to this client keyring class as its keyring. You are recommended to do this using the System manager user interface, by completing the following steps:
   a. Expand the Configuration that contains the Client Style model
   b. On the pop-up menu of the Client Style model, click **Edit**, to display the Object Editor
   c. Change the value of the **SSL Key Ring File** attribute to **CbDevTestClKeyring**

d. To apply the change and close the Object Editor window, click the **OK** button

e. On the pop-up menu of the Configuration, click **Activate** to activate that configuration and create or update the client style properties file

f. Copy the client style properties file to your client host or web server

When you start your Java client you should specify this client style properties for it to use.

**Related Concepts**

"SSL and Certificates" on page 277
"Certificate Authorities" on page 280
"Models of Trust Validation" on page 282
"Securing your Enterprise" on page 36
"Security" on page 263

**Related Tasks**

"Create your own Self-Signed Test Certificate"
"Get a Test Certificate from a Certificate Authority" on page 303
"Plan for Signed Production and Test Certificates" on page 293
"Get a Production Certificate from a Certificate Authority" on page 305

## Create your own Self-Signed Test Certificate

Use this procedure to create your own self-signed test certificate for use when testing a server and its SSL-based Java clients.

You may need to do this because the Component Broker test certificate has expired, or because you want a self-signed test certificate that specifically recognizes your server.

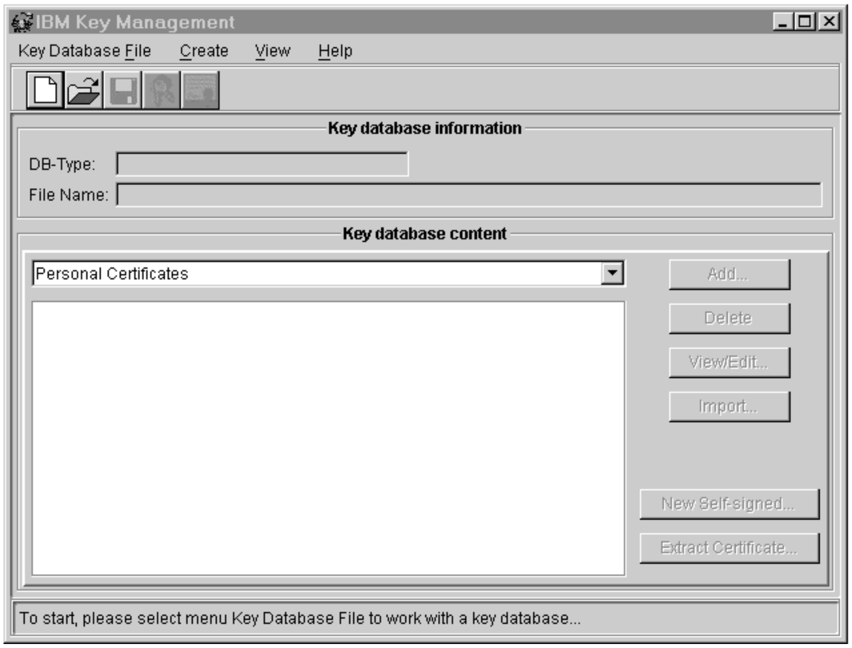
If you need a test certificate that has been signed by a Certificate Authority, use the procedure described in "Get a Test Certificate from a Certificate Authority" on page 303.

**To create your own self-signed test certificate, complete the following steps:**

1. Start the IKMGUI user interface, either by selecting the Server Key Management start icon or by typing **ikmgui** at a command prompt.

   WIN Click **Start - Programs - IBM Component Broker for Windows NT -Server Key Management**

   This displays the IBM Key Management window shown in Figure 52 on page 297. If you want to change the visible properties of this window, you can switch between **Metal**, **Windows**, and **Motif** under the **View** pull-down.



Personal Certificates

To start, please select menu Key Database File to work with a key database...

*Figure 52. The IBM Key Management window (IKMGUI)*

2. Open a new key database file. To do this, either click the **Create a new key database file** button or select **Key Database File - New** from the menu bar.

   You are then prompted to enter the file name for the key database.
3. Specify a unique file name for the key database, typically in the following form:

   ```
   <server_name>.kdb
   ```

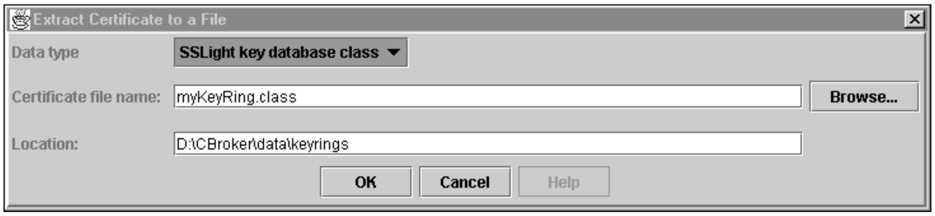
   Where <server_name> is the name of the server for which you are creating the key database.

   You are then prompted to enter a password for the key database.
4. Specify **Cbroker** as the password.

   Do not set the check-box for an expiration time for the password or to stash the password to a file; these just make things unnecessarily complicated.
5. After specifying the password for the keyring, the Server Key Management tool displays all of the default signer certificates. You can add, view or delete signer certificates from this screen. To continue creating a self-signed certificate, either click the **Create a new self-signed certificate** button or select **Create - New Self-Signed Certificate...** from the menu bar. You are then prompted for the certificate attributes.
6. Fill in the certificate attributes, including the name of your server as the distinguished name.
7. If you have only one personal certificate it will be set as the default certificate for the database. If you have more than one personal certificate, you will have to choose which one is the default certificate. You can change the default certificate by first highlighting the certificate and then selecting **View/Edit...**. In the following screen, select the checkbox at the bottom to set this certificate as the default.

   You should next export the new server certificate into a client-side keyring class.
8. Click the **Extract Certificate** button to export the certificate.

You are then prompted for the type, name, and location of the certificate file to produce, as shown in Figure 53.



Extract Certificate to a File

Data type: SSLight key database class

Certificate file name: myKeyRing.class

Location: D:\CBroker\data\keyrings

*Figure 53. Extract Certificate to a File window*

9. Click the **Data type** pull-down button and select the **SSLight key database class** option. This changes the file name extension to .class.
10. Type a file name for your keyring class. In the example shown in Figure 53, the file is called *myKeyRing.class*.

    Leave the location to default to the Component Broker keyrings directory. WIN by default, the Component Broker keyrings directory is *c:\Cbroker\data\keyrings*.
11. To create the keyring class, click the **OK** button.

    At this point, both the original CMS key database and the client-side keyring class exist in the Component Broker keyrings directory of the host on which you ran IKMGUI.
12. Close the IBM Key Management window to exit IKMGUI.

You must ensure that the server keyring file (the CMS Key Database) is copied to the Component Broker keyrings directory on the server host, and that the client keyring class file is copied to the URL location specified in the Client Style properties file of your Java client. For more information, see “Place Server and Client Keyrings in your Enterprise” on page 311.

### Related Concepts

“SSL and Certificates” on page 277
“Certificate Authorities” on page 280
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

### Related Tasks

Add a Server Certificate to a Client Keyring Class
“Use the Test Certificate Provided with Component Broker” on page 295
“Get a Test Certificate from a Certificate Authority” on page 303
“Plan for Signed Production and Test Certificates” on page 293
“Get a Production Certificate from a Certificate Authority” on page 305
“Place Server and Client Keyrings in your Enterprise” on page 311

## Create a Certificate Signing Request and Server Keyring File

Use this procedure to create a *Certificate Signing Request (CSR)*, which you will need to send to a certificate authority (CA) to get a signed certificate for a server from the CA. You only need to complete this procedure if you want to get a signed test or production certificate from a CA.

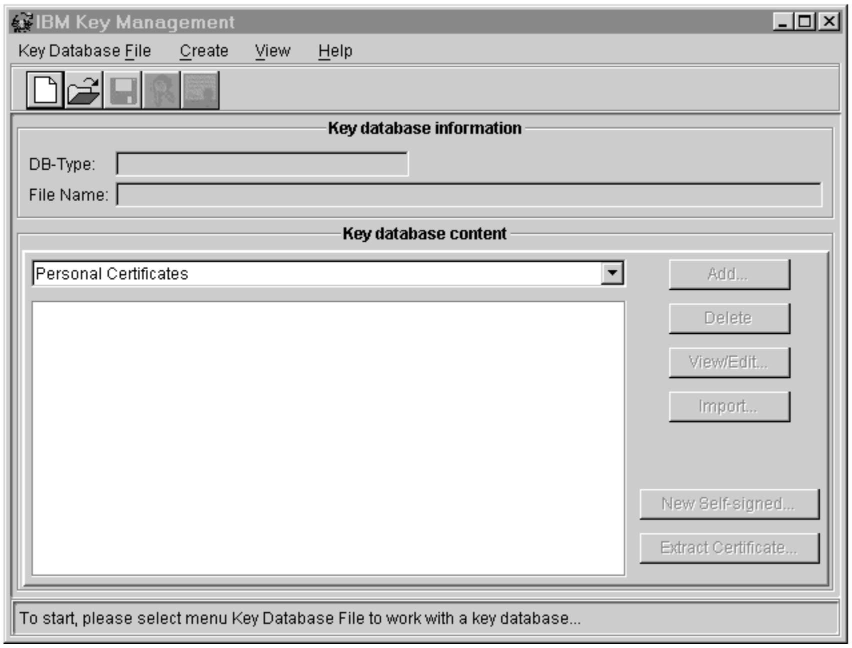
This procedure creates the CSR file in the Component Broker keyring directory. It also automatically creates a corresponding private key for the server that remains in your keyring database. You do not transmit the certificate's private key to the CA, and thus the private key remains entirely in your possession at all times.

**To create Certificate Signing Request (CSR), complete the following steps:**

1. Start the IKMGUI user interface, either by selecting the Server Key Management start icon or by typing **ikmgui** at a command prompt.

   WIN Click **Start - Programs - IBM Component Broker for Windows NT -Server Key Management**

   This displays the IBM Key Management window shown in Figure 54. If you want to change the visible properties of this window, you can switch between **Metal**, **Windows**, and **Motif** under the **View** pull-down.



To start, please select menu Key Database File to work with a key database...

*Figure 54. The IBM Key Management window (IKMGUI)*

2. Open a new key database file. To do this, either click the **Create a new key database file** button or select **Key Database File - New** from the menu bar.

   You are then prompted to enter the file name for the key database.

3. Specify a unique file name for the key database, typically in the following form:

   ```
   <server_name>.kdb
   ```

   Where <server_name> is the name of the server for which you are creating the CSR (and keyring).

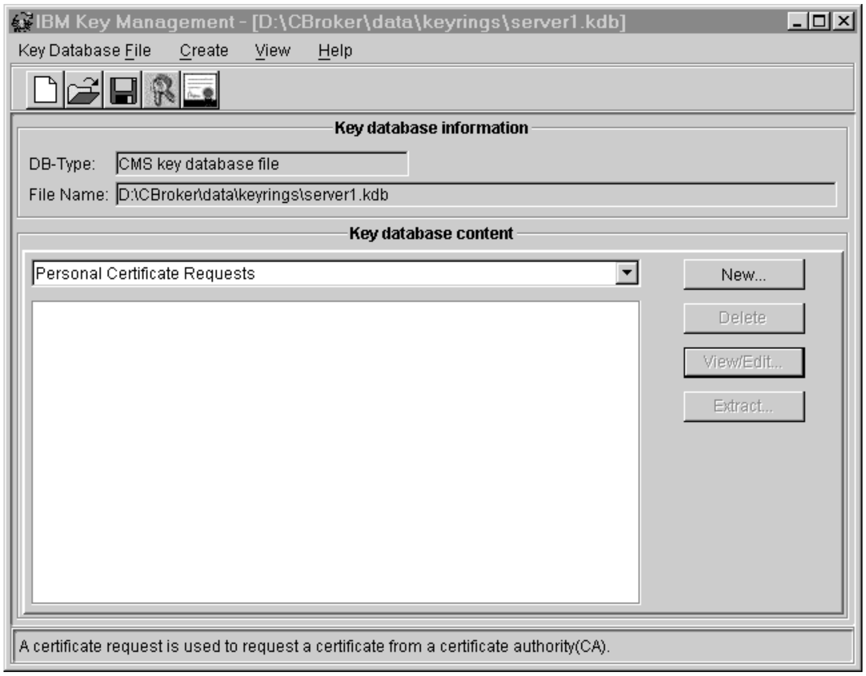
You are then prompted to enter a password for the key database.

4. Specify **Cbroker** as the password. If you use any other password you must specify that password on the **SSL Key Ring Password** attribute of the Server model, then activate the Configuration that contains that Server model to implement the password.

   Do not set the check-box for an expiration time for the password or to stash the password to a file; these just make things unnecessarily complicated.

5. Select the pull-down under **Key database content** then Scroll the pull-down list and select **Personal Certificate Requests**.

   This updates the IBM Key Management window, as shown in Figure 55.



DB-Type: CMS key database file

File Name: D:\CBroker\data\keyrings\server1.kdb

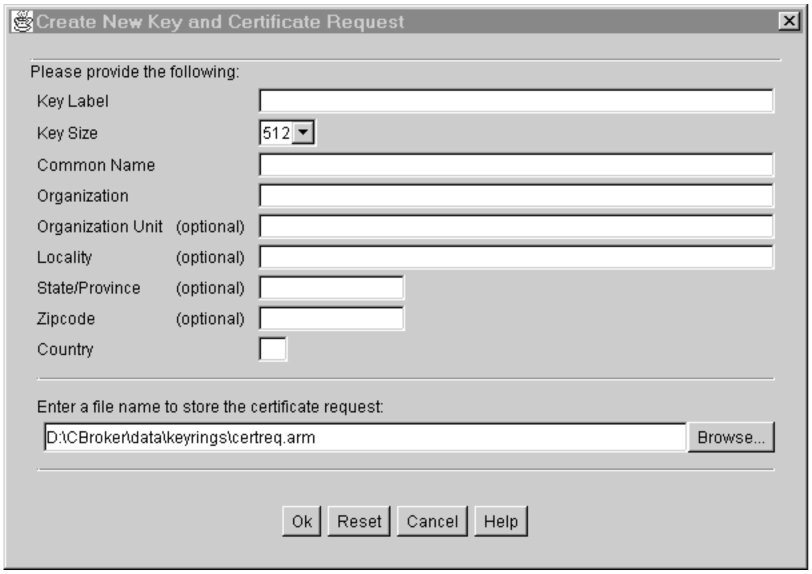
Personal Certificate Requests

A certificate request is used to request a certificate from a certificate authority(CA).

*Figure 55. The IKMGUI window, showing key database information*

6. Click the **New...** button.

   You are then prompted for the certificate attributes, as shown in Figure 56 on page 301.



Create New Key and Certificate Request

Please provide the following:

Key Label

Key Size 512

Common Name

Organization

Organization Unit (optional)

Locality (optional)

State/Province (optional)

Zipcode (optional)

Country

Enter a file name to store the certificate request:

D:\CBroker\data\keyrings\certreq.arm

*Figure 56. The Create New Key and Certificate Request window*

7. Fill in the following certificate attributes. (You can leave other attributes with their default values.)

**Key Label**

Give the certificate a key label, which is used to uniquely identify the certificate within the keyring. In Component Broker you typically have only one certificate in each server keyring, so you can assign any label value. However, it is good practise to use a unique label, perhaps related to the server name.

**Common Name**

Type the common name for the certificate. This is the primary, universal identity for the Certificate; that is, it should uniquely identify the principal that it represents. Since, in a Component Broker environment, certificates are used to represent server principals, the common convention is to use the form <host_name>/<server_name>. For example, for the server **PolicyServer1** on the host **centralops.xyz.com** you would set the common name to **centralops.xyz.com/PolicyServer1**.

**Note:** For some CAs, including the fully-qualified name of your host in the common name is more than just a convention. For example, VeriSign will not sign your certificate unless the domain portion of the host name is owned by your organization. So when preparing to set the common name, check the format that your CA requires.

**Organization ... Country**

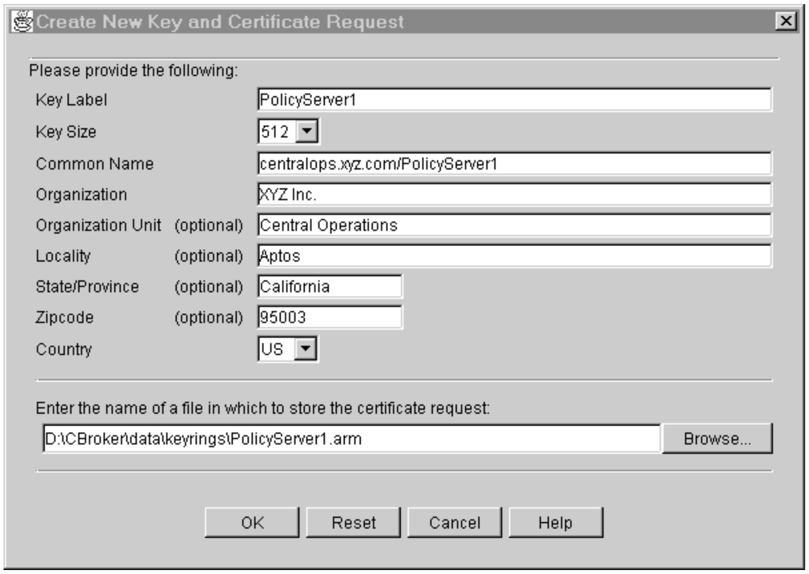
Type the name of your organization (for example, My Own Inc.), organization unit (for example, Central Operations), location (city), state/province (if applicable), zipcode (if applicable), and the two-letter identifier of the country in which the server belongs.

**Note:** *Some certificate authorities, including VeriSign, require that you spell out completely the state or province. For example, you need to specify California as opposed to CA.*

**The name of the file in which to store the certificate request**

Type the name of the file in which you want to store the CSR; typically something like the name of the server along with the .arm file extension, and stored in the keyrings directory.

This updates the Create New Key and Certificate Request window, as shown in Figure 57.



Create New Key and Certificate Request

Please provide the following:

| | | |
|---|---|---|
| Key Label | | PolicyServer1 |
| Key Size | | 512 |
| Common Name | | centralops.xyz.com/PolicyServer1 |
| Organization | | XYZ Inc. |
| Organization Unit | (optional) | Central Operations |
| Locality | (optional) | Aptos |
| State/Province | (optional) | California |
| Zipcode | (optional) | 95003 |
| Country | | US |

Enter the name of a file in which to store the certificate request:

D:\CBroker\data\keyrings\PolicyServer1.arm

*Figure 57. The Create New Key and Certificate Request window, with example details*

8. When you have filled in all of the required fields for the certificate, click the **OK** button.
9. When the CSR file has been created, you will be notified and prompted to get the certificate signed.

   *be sure to set the signed certificate as the default for the server keyring.*

After completing this procedure to create a CSR, you should send the CSR to your CA to get a signed certificate, by using one of the following procedures:

- “Get a Test Certificate from a Certificate Authority” on page 303
- “Get a Production Certificate from a Certificate Authority” on page 305

### Related Concepts

“SSL and Certificates” on page 277
“Certificate Authorities” on page 280
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

### Related Tasks

“Add a Server Certificate to a Client Keyring Class” on page 310
“Use the Test Certificate Provided with Component Broker” on page 295
“Get a Test Certificate from a Certificate Authority” on page 303
“Plan for Signed Production and Test Certificates” on page 293
“Get a Production Certificate from a Certificate Authority” on page 305

## Get a Test Certificate from a Certificate Authority

Use this procedure to get a test certificate for a server from a certificate authority (CA). You do this by sending a Certificate Signing Request (CSR) to the CA.

You are recommended to use this procedure to obtain a test certificate at least once before getting a production certificate for your server, as described in “Get a Production Certificate from a Certificate Authority” on page 305.

This procedure is based on use of VeriSign as the CA, which offers a test certificate for free. The test certificate is a legitimate certificate, fully signed and endorsed for actual use, and so the procedure described can be used to validate that you have everything set up right. However, the test certificate expires in two weeks from receipt and so is not useful for production use.

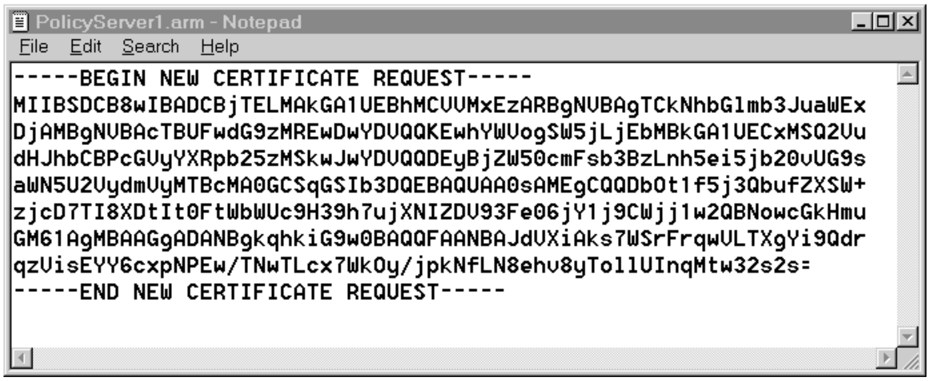
**Prerequisites:** You must have already created the Certificate Signing Request (CSR) needed, as described in “Create a Certificate Signing Request and Server Keyring File” on page 299.

**To get a test certificate for a server from a certificate authority (CA), complete the following steps:**

1. Start your Web browser and link to VeriSign’s home page at http://www.verisign.com.
2. Click **Server IDs** to request that your certificate get signed for a server.
3. Click **Test Drive a Trial Server ID for Free!**. This displays the **VeriSign Test Digital ID Enrollment** page.

   **Note:** be sure to read the information comparing Test Digital IDs to Secure Server Digital IDs.
4. Follow the instructions to enroll in the Secure Server Test Drive. When you get to the page to submit the CSR, scroll down to the edit box where you will enter the CSR you created in the previous section.
5. Open the .arm file containing the CSR that you created in “Create a Certificate Signing Request and Server Keyring File” on page 299. In the example in that procedure, the CSR was saved in the file **PolicyServer1.arm** in the Component Broker keyrings directory. You can open this file with any text editor that supports cut-and-paste actions.

   WIN You can use the Windows NT **Notepad** editor.

   ```
   -----BEGIN NEW CERTIFICATE REQUEST-----
   MIIBSDCB8wIBADCBjTELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEx
   DjAMBgNVBAcTBUFwdG9zMREwDwYDVQQKEwhYWVogSW5jLjEbMBkGA1UECxMSQ2Vu
   dHJhbCBPcGVyYXRpb25zMSkwJwYDVQQDEyBjZW50cmFsb3BzLnh5ei5jb20vUG9s
   aWN5U2VydmVyMTBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDbOt1f5j3QbufZXSW+
   zjcD7TI8XDtIt0FtWbWUc9H39h7ujXNIZDV93Fe06jY1j9CWjj1w2QBNowcGkHmu
   GM61AgMBAAGgADANBgkqhkiG9w0BAQQFAANBAJdVXiAks7WSrFrqwULTXgYi9Qdr
   qzVisEYY6cxpNPEw/TNwTLcx7WkOy/jpkNfLN8ehv8yTollUInqMtw32s2s=
   -----END NEW CERTIFICATE REQUEST-----
   ```
6. In your editor window, select all of the text, then copy the selected text, including the header

   ——-BEGIN NEW CERTIFICATE REQUEST——-

and the corresponding trailer.

7. Return to the Enroll page in your Web browser, then paste the text into the edit box. For example, ensure the cursor in the edit box, then click on **Edit - Paste**.
8. Click **CONTINUE**.

   This display a verification page with the following three steps:

   a. **Verify Your Distinguished Name:**
   Verify all of the information displayed about your certificate. In particular, ensure that the Common Name is correct and unique.

   b. **Enter Your Contact Information:**
   Enter the requested information about you. VeriSign needs this information to send you your signed certificate. In particular, make sure your e-mail address is correct. (VeriSign will e-mail your signed certificate to that e-mail address.)

   c. **Read the Digital ID Subscriber Agreement:**
   Read the terms and conditions stipulated by VeriSign about the Test ID you are requesting.

   *IF YOU DO NOT ACCEPT THESE CONDITIONS, DO NOT CONTINUE.*
9. When complete, and if you accept the VeriSign Subscriber Agreement, click **Accept**.
10. You should get an acknowledgement indicating that you have successfully completed your request. The page goes on to suggest that you need to download and install the Verisign test certificate in your Web browser.

    *DO NOT PROCEED TO INSTALL THIS CERTIFICATE INTO YOUR WEB bROWSER AS DESCRIBED bY VERISIGN.*

    The Verisign test certificate should be included in the keyring used by Component Broker, not your Web browser. Fortunately, this certificate was already included in your new keyring file when you created it using the Server Key Management tool.

    If you are using some other Certificate Authority, you can check to see whether the certificate of that Certificate Authority is already included in your keyring file by selecting the pull-down under **Key database content** and selecting **Signer Certificates**. If the CA's certificate is not in the keyring, then you should follow the CA's procedure for obtaining the certificate. Typically, you will want to receive the certificate as a file and use the **Add** button on the **Signer Certificates** screen to add the certificate to your keyring.

    You will need their Test CA certificate, but you need to define that certificate to Component Broker, not your Web browser. This involves a slightly awkward series of steps, but this is necessary to successfully install the test certificate you will receive back from VeriSign.

### Related Concepts

"SSL and Certificates" on page 277
"Certificate Authorities" on page 280
"Models of Trust Validation" on page 282
"Securing your Enterprise" on page 36
"Security" on page 263

### Related Tasks

"Add a Server Certificate to a Client Keyring Class" on page 310
"Use the Test Certificate Provided with Component Broker" on page 295
"Create a Certificate Signing Request and Server Keyring File" on page 299

“Plan for Signed Production and Test Certificates” on page 293
“Get a Production Certificate from a Certificate Authority”

## Get a Production Certificate from a Certificate Authority

Use this procedure to get a production certificate for a server from a certificate authority (CA). You do this by sending a Certificate Signing Request (CSR) to the CA.

This procedure is based on use of VeriSign as the CA.

**Prerequisites:**

- You must have already created the Certificate Signing Request (CSR) needed, as described in “Create a Certificate Signing Request and Server Keyring File” on page 299.
- Getting any commercial CA to sign your certificates can be expensive. For example, VeriSign charges from around $349.00 up to around $1000.00, depending on the type of certificate you want signed and the relative strength you need. In some cases, the CA offers volume discount plans, but the cost is sufficient that you will not want to waste any of the certificates you get signed.
- You are recommended to get a test certificate for your server from the CA at least once before getting a production certificate, as described in “Get a Test Certificate from a Certificate Authority” on page 303. The test certificate is a legitimate certificate, fully signed and endorsed for actual use, and so the procedure described can be used to validate that you have everything set up right.

**To get a production certificate for a server from a certificate authority (CA), complete the following steps:**

1. Start your Web Browser and link to VeriSign’s home page at http://www.verisign.com.
2. Click **Server IDs** to request that your certificate get signed for a server.
3. Click **Get Your Secure Server ID Now!** to request that your certificate get signed for a server.
4. You are then led through the following series of pages that exchange the information you need to know with the information VeriSign needs to process your certificate request.

   After completing a page, display the next page by clicking the **Continue** button at the bottom of the page.
5. The **Before You Start** page lists the things you should do before beginning this process, including installing your ″Web Server Software″, setting up your Internet proxies, determining how you will pay for the certificate, reviewing the legal agreement and, if necessary, printing the enrollment guide. In your case, you should treat any references to ″Web Server Software″ to mean the CBConnector software.
6. The **Step 1: Confirm Domain Name** page informs you that you (your enterprise) must own the domain name indicated in the common name of your certificate. (These domain names are registered with NIC.) For example, if you had specified the common name **centralops.xyz.com/PolicyServer1**, then xyz.com would be considered a domain name. VeriSign will verify that the domain name you specified belongs to your enterprise; this is part of the authentication process completed by Certificate Authorities.

7. The **Step 2: Obtain Proof of Right** page provides instructions on another authentication step that VeriSign performs. In this case, you must prove that your enterprise has the right to operate under the Organization Name that you specified in your CSR. The VeriSign process is optimized to using D-U-N-S numbers for this purpose. If you take this approach, you either need to get your D-U-N-S number, or if you are a U.S. company VeriSign can look that up for you.

   If you don't have a D-U-N-S number, or if you don't want to use this to prove your right to the Organization name, then you can provide alternate proof of right. VeriSign does not describe what this might be, but if you have a letter of Incorporation or a similar article, you can fax that to VeriSign as proof. The process will be a little slower in this case, because you will not be able to proceed until VeriSign has received and processed the alternative proof (fax).
8. The **Step 3: Generate CSR** page instructs you to create your CSR. You should have already done this, as described in "Create a Certificate Signing Request and Server Keyring File" on page 299.
9. The **Step 4: Submit CSR** page provides you with an edit box to paste your CSR.

   As with creating a Test certificate, use a graphical text editor (such as Notepad on Windows NT) to open your CSR .arm file (for example, PolicyServer1.arm). Select and copy all the text in the CSR file, including the header

   ```
   —–-BEGIN NEW CERTIFICATE REQUEST—–-
   ```

   and the corresponding trailer.

   Paste the text into the edit box on the Submit CSR document.
10. The **Step 5: Complete Application** page is perhaps the most time-consuming and difficult step in this process. On this page you verify your distinguished name, and enter the following information:

    **Your server information**

    You are asked to select your server software vendor. Click the pull-down button and select IBM.

    You are also asked to supply a Challenge Phrase, which can be any text string you choose. However, you should treat it like a password. You will be asked to present this same challenge phrase if you submit a renewal request, or if you request to revoke your certificate (for example, if your certificate is compromised). You may also be asked to supply this challenge phrase when speaking with VeriSign.

    **Your technical contact information.**

    This should identify you. Your e-mail address is particularly important here as this is where the signed certificate will be sent.

    **Your organizational contact information**

    This should be someone, other than yourself, who is a member of your enterprise. VeriSign will contact them during the authentication process to verify the legitimacy of your request.

    **Your billing contact information**

    This can be you or someone else in your organization who is responsible for payment.

    **The type of Secure Server ID that you are requesting**

    **Your payment information**

**Your organizational information (that is, your D-U-N-S number)**
This is the D-U-N-S number of your organization. If you use an alternate proof of right, then VeriSign will instruct you on how to fill out this information.

11. Review the Server Certificate Agreement.

    If you do not accept the conditions of this agreement, then click **DECLINE**. Otherwise, to submit your request click **ACCEPT**.
12. VeriSign will send you an e-mail containing your signed production certificate.

When you have received your signed certificate you can go on to retrieve and import it into your keyring, as described in "Receive a Signed Certificate into a Server Keyring".

**Related Concepts**

"SSL and Certificates" on page 277
"Certificate Authorities" on page 280
"Models of Trust Validation" on page 282
"Securing your Enterprise" on page 36
"Security" on page 263

**Related Tasks**

"Plan for Signed Production and Test Certificates" on page 293
"Create a Certificate Signing Request and Server Keyring File" on page 299
"Receive a Signed Certificate into a Server Keyring"
"Add a Server Certificate to a Client Keyring Class" on page 310
"Get a Test Certificate from a Certificate Authority" on page 303
"Use the Test Certificate Provided with Component Broker" on page 295

## Receive a Signed Certificate into a Server Keyring

Use this procedure to receive a signed certificate from a certificate authority (CA) into your server keyring. You only need to complete this procedure if your CA has sent you a signed test or production certificate that you want to use with Component Broker.

This procedure can be used in the same way for both test certificates and production certificates. The primary difference between the two is the amount of time it takes for the CA to authenticate the principal your certificate represents. Test certificates are authenticated automatically based on some simple edit checks and should be returned to you in a matter of minutes or a few hours. Production certificates may take several days or a week to authenticate and return to you.

This procedure is based on use of VeriSign as the CA. The e-mail you get back from VeriSign contains text identifying the common name of the server you specified in your certification request (CSR). So you should be able to match it up to the appropriate keyring if you have used a consistent naming scheme; typically, by giving the keyring database file and the certificate within it the same relative name.

**Prerequisites:**

- You must have already requested a signed certificate, as described in "Get a Production Certificate from a Certificate Authority" on page 305 or "Get a Test Certificate from a Certificate Authority" on page 303.

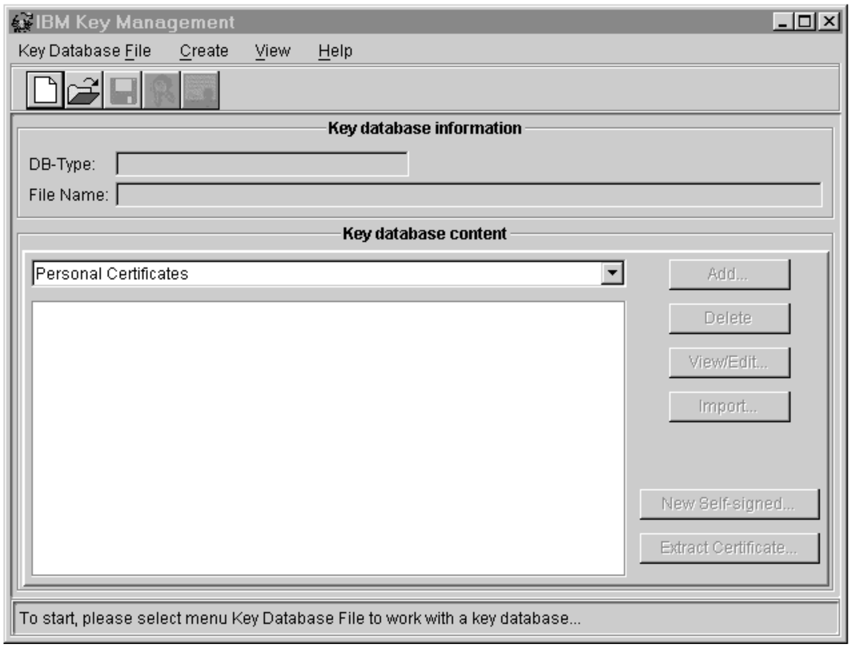
- You must have already have received an e-mail containing the signed certificate from the CA.

**To receive a signed certificate into your server keyring, complete the following steps:**

1. When you receive e-mail from VeriSign containing your test or production certificate, save that mail in a file. In this example, the certificate was saved into **PolicyServer1.responseMail.arm**.
2. Start the IKMGUI user interface, either by selecting the Server Key Management start icon or by typing **ikmgui** at a command prompt.

   WIN Click **Start - Programs - IBM Component Broker for Windows NT -Server Key Management**

   This displays the IBM Key Management window shown in Figure 58. If you want to change the visible properties of this window, you can switch between **Metal**, **Windows**, and **Motif** under the **View** pull-down.



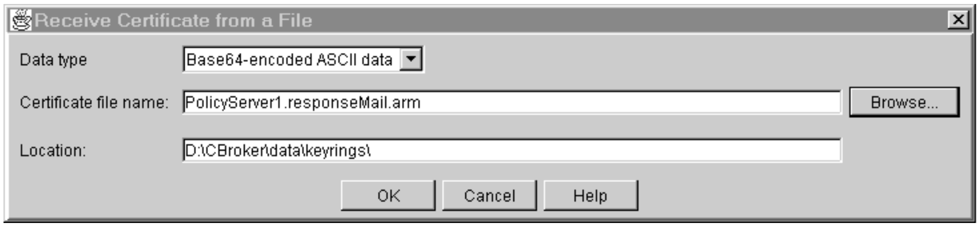
To start, please select menu Key Database File to work with a key database...

*Figure 58. The IBM Key Management window (IKMGUI)*

3. Open an existing keyring by either clicking on the **Open a key database file...** button or by selecting **Key Database File - Open** from the menu bar. Type the name and location of the keyring at the prompt, then click the **OK** button. Enter the password at the prompt, then click the **OK** button.
4. Click on the certificate types pull-down button beneath **Key Database Context** and select **Personal Certificates** (the default).
5. To receive your signed certificate into the keyring, click the **Receive...** button.

   This displays the **Receive Certificate from a File** dialog window.
6. In the Receive Certificate from a File dialog window, type the name of the file containing your e-mail response from VeriSign. If needed, use the **browse...** to find and select the e-mail file that you saved.

   For example, see Figure 59 on page 309.



Data type: Base64-encoded ASCII data

Certificate file name: PolicyServer1.responseMail.arm

Location: D:\CBroker\data\keyrings\

*Figure 59. The Receive Certificate from a File dialog window*

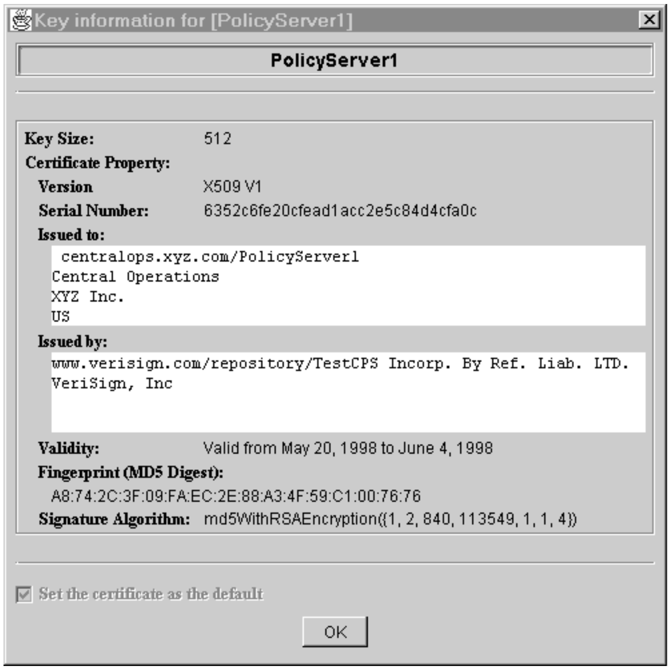
7. To receive your certificate, click the **OK** button.

   If you are installing a test certificate, and the Verisign test certificate is not in your keyring, you will get an error message at this point. When you create a new keyring using the Server Key Management tool, the Verisign test certificate should have already been added to the keyring along with certificates from several other Certificate Authorities (CA).

   You might encounter the same error if you are installing a production certificate signed by an unknown CA. If so, use that CA's procedure for installing the signed certificate in your keyring.

8. Optionally, to verify the certificate, click the **View/Edit...** button in the main IBM Key Management window.

   This displays information like that shown in the following figure:


Key information for [PolicyServer1]
PolicyServer1
☑ Set the certificate as the default
OK

Key Size: 512

Certificate Property:

Version: X509 V1

Serial Number: 6352c6fe20cfead1acc2e5c84d4cfa0c

Issued to:

```
 centralops.xyz.com/PolicyServer1
Central Operations
XYZ Inc.
US
```

Issued by:

```
www.verisign.com/repository/TestCPS Incorp. By Ref. Liab. LTD.
VeriSign, Inc
```

Validity: Valid from May 20, 1998 to June 4, 1998

Fingerprint (MD5 Digest):
A8:74:2C:3F:09:FA:EC:2E:88:A3:4F:59:C1:00:76:76

Signature Algorithm: md5WithRSAEncryption({1, 2, 840, 113549, 1, 1, 4})

*Figure 60. Key Information for Certificate window*

When you have received the signed certificate for a server into its keyring, as described in this procedure, you can transfer the public portion of that certificate into the client keyrings used by clients that communicate with the server. This is described in the procedure "Add a Server Certificate to a Client Keyring Class" on page 310.

### Related Concepts

"SSL and Certificates" on page 277

“Certificate Authorities” on page 280
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

“Plan for Signed Production and Test Certificates” on page 293
“Create a Certificate Signing Request and Server Keyring File” on page 299
“Get a Production Certificate from a Certificate Authority” on page 305
“Add a Server Certificate to a Client Keyring Class”
“Get a Test Certificate from a Certificate Authority” on page 303
“Use the Test Certificate Provided with Component Broker” on page 295

## Add a Server Certificate to a Client Keyring Class

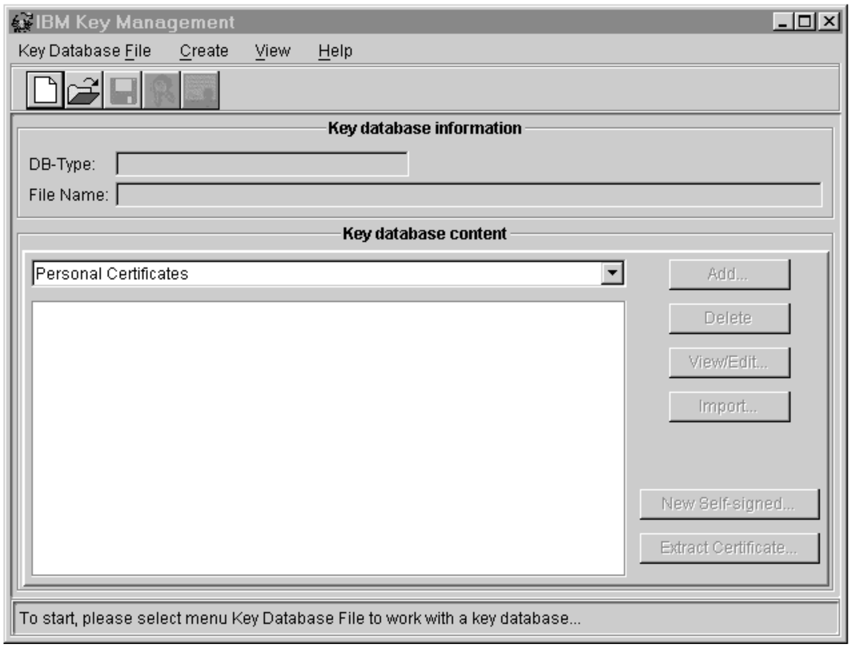
Use this procedure to add a server certificate to a client keyring. You only need to complete this procedure if you want to build a client-side keyring that includes the server certtificate.

**To add a server certificate to a client keyring, complete the following steps:**

1. Start the IKMGUI user interface, either by selecting the Server Key Management start icon or by typing **ikmgui** at a command prompt.

   WIN Click **Start - Programs - IbM Component Broker for Windows NT -Server Key Management**

   This displays the IBM Key Management window shown in Figure 61. If you want to change the visible properties of this window, you can switch between **Metal**, **Windows**, and **Motif** under the **View** pull-down.

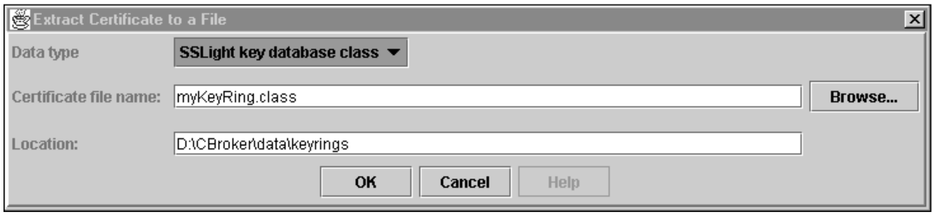


To start, please select menu Key Database File to work with a key database...

*Figure 61. The IBM Key Management window (IKMGUI)*

2. Open the server’s key database file. To do this, either click the Open File icon or select **Key Database File - Open** from the menu bar, and specify the name of the file required.
3. Click the **Extract Certificate** button to export the certificate.

   You are then prompted for the type, name, and location of the certificate file to produce, as shown in Figure 62.



Extract Certificate to a File

Data type: SSLight key database class

Certificate file name: myKeyRing.class

Location: D:\CBroker\data\keyrings

*Figure 62. Add Certificate to a File window*

4. Click the **Data type** pull-down button and select the **SSLight key database class** option. This changes the file extension to .class.
5. Type a name for your keyring class file or choose an existing class file using the **browse** button. If this is a new keyring class file, choose an appropriate location.

   WIN by default, the Component Broker keyrings directory is *c:\Cbroker\data\keyrings*.
6. To create the certificate file, click the **Ok** button.

   To create the keyring class file, click OK. At this point, the client keyring class should contain the server’s certificate.
7. Close the IBM Key Management window to exit IKMGUI.

**Related Concepts**

“SSL and Certificates” on page 277
“Certificate Authorities” on page 280
“Models of Trust Validation” on page 282
“Keyrings” on page 284
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

“Use the Test Certificate Provided with Component Broker” on page 295
“Get a Test Certificate from a Certificate Authority” on page 303
“Plan for Signed Production and Test Certificates” on page 293
“Get a Production Certificate from a Certificate Authority” on page 305
“Place Server and Client Keyrings in your Enterprise”

## Place Server and Client Keyrings in your Enterprise

Use this procedure to place server and client keyrings onto appropriate hosts in your enterprise. You need to do this after you have created a new keyring, or have added a new certificate to a keyring.

To place server and client keyrings in your enterprise you must manually copy those keyring files into the Component Broker keyrings directory on each of the hosts where you will need them. (Distribution of keyrings throughout your enterprise is not currently automated through system management.)

You must copy a server keyring to the Component Broker keyrings directory on the server host. Likewise, you must copy the client keyring to one of the following locations:

- For Java application clients, in the Component Broker keyrings directory on the client host
- For Java applet clients, on the Web Server containing the HTML web pages and Java classes that comprise the Java applet and the Component Broker runtime, including the embedded SSLight libraries.

When you install Component Broker on a host, it automatically creates a distinct directory specifically for storing keyrings. This directory is known as the Component Broker *keyrings directory*.

WIN by default, the Component Broker keyrings directory is created in **c:\Cbroker\data\keyrings**, depending on the drive and base directory you specified when installing Component Broker.

The same keyrings directory is created for both client hosts and server hosts where you install Component Broker. If you follow the procedures for installing Component Broker at a Web Server, as provided in the *Quick Beginnings Guide*, a similar directory is created on your Web Server.

When you have copied the keyring files onto the hosts that need them, you must configure the corresponding Client Style and Server models in your system management Configuration to refer to their respective keyrings. For more information about this, see “Enable Security within a Configuration” on page 321.

### Related Concepts

“SSL and Certificates” on page 277
“Certificate Authorities” on page 280
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

### Related Tasks

“Plan for Signed Production and Test Certificates” on page 293
“Create a Certificate Signing Request and Server Keyring File” on page 299
“Get a Production Certificate from a Certificate Authority” on page 305
“Add a Server Certificate to a Client Keyring Class” on page 310
“Get a Test Certificate from a Certificate Authority” on page 303
“Use the Test Certificate Provided with Component Broker” on page 295
“Receive a Signed Certificate into a Server Keyring” on page 307

## Create and Protect Server Keytab Files

Use the following procedures to create and protect keytab files for application servers:

- “Protect Server Keytab Files” on page 315

Use this procedure to protect the keytab file used by an application server from unauthorized tampering. You should do this for all keytab files used by application servers, including the default keytab file created automatically by Component Broker for each server host.

- “Create a Unique Keytab File for an Application Server”

  Use this procedure to create a unique keytab file for an application server. You only need to complete this procedure if you do not want the server to use the default keytab file created automatically by Component Broker for the server host.

**Related Concepts**

“Securing your Enterprise” on page 36
“Security” on page 263
“Use of DCE- and SSL-Based Security in Your Enterprise” on page 271
“DCE and the User Registry” on page 274

**Related Tasks**

“Enable Security within a Configuration” on page 321
“Create and Install Server Certificates” on page 292
“Change the Password for a Server Principal” on page 320
“Administer Accounts for Client and Server Principals” on page 316
“Configure Security for a Server” on page 323

## Create a Unique Keytab File for an Application Server

Use this procedure to create a unique keytab file for an application server. You only need to complete this procedure if you do not want the server to use the default keytab file created automatically by Component Broker for the server host.

Component Broker automatically creates a single default keytab file for each host, and stores in that file the keytab entries for all servers on that host, as follows:

- WIN c:\CBroker\data\v5cbctab
- AIX /usr/lpp/CBroker/data/v5cbctab

The server principal information for every server on that host is entered into that keytab file. Thus, if different servers on the host are started under different local operating system identities, then each of these identities will have to be enabled to access that keytab file.

This may unacceptable if you have some servers that are started manually by different administrators and you do not want them to all have access to the default keytab file.

You can change the keytab file for a given server by creating your own keytab file and specifying the new filename or location in the **keytab file name** attribute of the corresponding Server model. Component Broker will use this file to store and retrieve the login information for the server.

**Prerequisites:**

As part of this procedure, you must supply the principal name of the server for which an account should be created in the DCE user registry. You can use the

System Manager user interface to determine this name from the **principal name** attribute of the Server model for the server, by completing the following steps:

1. Expand the Configuration that contains the Server model.
2. Expand the Server Groups folder or Servers (freestanding) folder. If your server is a member of a server group, the principal name is set on the Server Group model; otherwise, it is set on the Server (freestanding) model.
3. On the pop-up menu of the Server Group model or Server (freestanding) model, click **Edit**, to display the Object Editor window.
4. Note the value of the **principal name** attribute. If this is not appropriate, change it to a suitable value.
5. To close the Object Editor window, and save any changes that you have made to attributes, click the **OK** button.

**To create a unique keytab file for a server, complete the following steps:**

1. Use the DCE administration tools (rgy_edit) to create a unique keytab file. In doing so, you must supply the name of the server for which an account should be created in the DCE user registry. The corresponding account for the server in the DCE user registry should be created automatically. For more information about how to use the DCE administration tools, see the information provided with the DCE product.
2. Change the **keytab file name** attribute of the Server model for the server to indicate the name of the keytab file so that Component Broker can find the keytab file to log in the server during server start up. To do this, complete the following steps:
   a. Expand the Configuration that contains the Server model.
   b. Expand the Server Groups folder or Servers (freestanding) folder. If your server is a member of a server group, the principal name is set on the Server Group model; otherwise, it is set on the Server (freestanding) model.
   c. On the pop-up menu of the Server Group model or Server (freestanding) model, click **Edit**, to display the Object Editor window.
   d. Note the value of the **keytab file name** attribute. If this is not appropriate, change it to a suitable value.
   e. To close the Object Editor window, and save any changes that you have made to attributes, click the **OK** button.

You should protect the server keytab file, as described in “Protect Server Keytab Files” on page 315.

### Related Concepts

"Client Platforms and Configurations" in the *Quick Beginnings* book
“Client Styles” on page 26
“Clients” on page 207
“Wizards” on page 447

### Related Tasks

“Protect Server Keytab Files” on page 315
“Create and Install Server Certificates” on page 292

## Protect Server Keytab Files

Use this procedure to protect the keytab file used by an application server from unauthorized tampering. You should do this for all keytab files used by application servers, including the default keytab file created automatically by Component Broker for each server host.

A keytab file can be used for authenticating a server without requiring a local administrator to log in for the server. Thus the keytab contains sensitive security information, specifically the server's userid and password. It is essential for the integrity of the server that this file be protected.

To protect a keytab file you set its file permissions so that only the servers that need to access the keytab file are authorized to do so.

**Prerequisites:**

- You need to know the operating system identity of each application server that is to use the keytab file. Most often, a server assumes the identity of the administrator registered for Component Broker with the host operating system ( WIN in the NT Registry). However, if the server, or the host's SM agent or ORB daemon, is started manually the server can assume the identity of another administrator. For more information about determining the operating system identity used by a server, see Server Keytab Files (page 276).
- The keytab file to be protected must already exist. If you do not want a server to use the default keytab file, called v5srvtab, you must create a unique keytab file for the server. For more information, see Server Keytab Files.
- WIN Files can only be protected if they are installed in the Windows NT file system (NTFS). For the appropriate procedure steps, see To protect a server keytab file on Windows NT (page 315)
- AIX DCE on AIX sets the permissions for keytab files automatically, so the keytab file should already have the correct permissions. If needed, change directory (**cd**) to /krb5, and use **chown** to set the owner to your administrator's id or root, and use **chmod** to set the permissions so that only the owner can read or write the keytab file.

**WIN To protect a server keytab file on Windows NT, complete the following steps:**

1. Log in to the host machine where the server resides, either as the administrator that normally starts the host machine, or the administrator that manually starts the part of the Component Broker whose local operating system identity is assumed by the server.
2. From the Windows NT Start menu, click **Programs - Windows NT Explorer**. This displays the Windows NT Explorer that you can use to display the icon for the keytab file that you want to protect.
3. In the Windows NT Explorer window, expand the directory into which you have installed DCE to display the contents of the **krb5** subdirectory that contains the DCE keytab files; for example, expand **C: - Opt - Digital - dcelocal - krb5**.
4. From the pop-up menu of the keytab file icon, click **Properties**.
5. In the Properties window, click the **Security** tab.

6. On the Security page, click the **Permissions** button. This displays the Permissions dialog window that you can use to change the file permissions for the keytab file.
7. Ensure only System and the group representing your administrator (that is, Administrators) has Full Control (All) permission. **Everyone** should have only Execute (X) permission. All other user groups and users should have No Access, or be removed from the ACL.
8. To accept the file permissions, click the **OK** button in the Permissions dialog window.
9. To close the Properties window, click its **OK** button.

### Related Concepts

"Securing your Enterprise" on page 36
"Security" on page 263

### Related Tasks

"Create a Unique Keytab File for an Application Server" on page 313
"Create and Install Server Certificates" on page 292
"Change the Password for a Server Principal" on page 320
"Administer Accounts for Client and Server Principals"
"Configure Security for a Server" on page 323

## Administer Accounts for Client and Server Principals

Use this set of procedures to create DCE accounts for all client principals that will use Component Broker, to change account passwords, and if needed to delete accounts.

- "Create an Account for a Client Principal" on page 317
- "Log in a Client Principal" on page 219
- "Change the Password for a Client Principal" on page 319
- "Change the Password for a Server Principal" on page 320

You must create a DCE account for each client principal that uses Component Broker.

Component Broker creates automatically an account for a new server principal, when you configure and activate a new server. However, you can change the keytab file for a server.

Before a principal can use Component Broker, it must log in. You log in a client principal manually, either by using an explicit dce_login command or in response to an automatic prompt from Component Broker. Servers, by default, log in automatically with the information in their keytab file.

It is essential to the security integrity of your enterprise that principals change their passwords from time to time. Client principals can use the rgy_edit command to change their own passwords. An administrator, with proper authority to access a server's keytab file, can use rgy_edit to change the server's password in that keytab file.

WIN You can use the DCE Director to delete accounts for client and server principals, as described in ″Using Digital DCE Director″ in the online DCE product information.

Generally, for the latest information about using DCE administration tools, and about managing principal (user) accounts, see the online DCE product information.

**Related Concepts**

"SSL and Certificates" on page 277<br>
"Certificate Authorities" on page 280<br>
"Accounts for Component Broker Administration" on page 287<br>
"Models of Trust Validation" on page 282<br>
"Securing your Enterprise" on page 36<br>
"Security" on page 263

**Related Tasks**

"Create and Install Server Certificates" on page 292<br>
"Create a Unique Keytab File for an Application Server" on page 313<br>
"Protect Server Keytab Files" on page 315<br>
"Chapter 11. Administer Security in your Enterprise" on page 261

## Create an Account for a Client Principal

Use this procedure to create a new DCE account for a client principal. You need to do this for every client principal that is to use Component Broker.

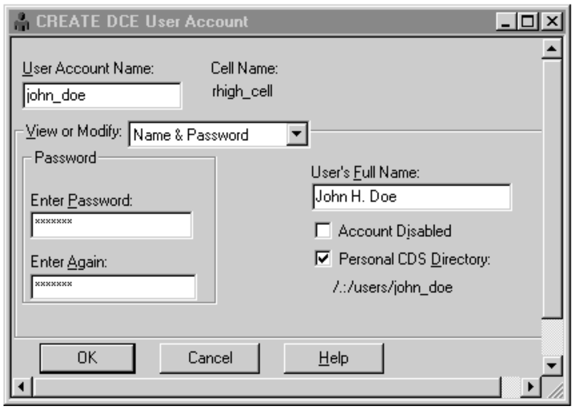
You can create an account for a client principal by using one of the following DCE administration tools:

- WIN The DCE Director or dcecp
- AIX dcecp

This procedure is based on using The DCE Director on Windows NT. For the latest information about using the DCE Director, see the DCE product documentation.

**To create a DCE account for a client principal, complete the following steps:**

1. Start the DCE Director; for example, by clicking on **Start - Programs - DCE for Windows NT V2.0 - DCE Director**
2. In the DCE Director window, double-click on the Users icon. This will open the DCE User Accounts window.
3. Click the **Create** icon, to display the **Create DCE User Account Dialog Window**, as shown in Figure 63 on page 318.



User Account Name: john_doe

Cell Name: rhigh_cell

View or Modify: Name & Password

Enter Password: *******

Enter Again: *******

User's Full Name: John H. Doe

□ Account Disabled

☑ Personal CDS Directory: /.:/users/john_doe

*Figure 63. The Create DCE User Account Dialog Window*

You will be prompted to enter the principal information for the account you are creating. Supply the account name (also referred to as a Security Name), an initial password, and the user's full name.

4. Specify a unique value for the **User Account Name**. (You will be warned if you select a name that is already being used for another account.) A common technique for selecting account names is to combine the principal's first and last name, separated by an underscore (_). If this turns out to conflict with another user, then add a middle initial, or perhaps an entire middle name, or reverse the first and last names to produce something unique. If all else fails, you might let the principal select their own arbitrary alias, such as a preferred nickname.
5. You must confirm the **Password** you selected by entering it twice. Generally, you select a randomly generated value for the initial password. Depending on the policy you have established for your Cell, principals are typically required to change their passwords the first time they log in.

   Bear in mind that principals that access Component Broker from Java clients by using SSL-based security are not able to change their passwords from their Java clients. You must either avoid a policy that requires that they change their password on first use, or provide another means for them to change their passwords remotely; perhaps by supplying a Web document accompanied by a CGI script to do this from your Web Server.
6. If you want to establish the group membership for this principal, click on the **View or Modify** pull-down button, then select **Group Memberships**. If needed, you can create new groups from the resulting window.

   You can also set UNIX account related information by selecting DCE UNIX Information from the same **View or Modify** pull-down menu.
7. After you entered the information for this account, click the **OK** button.
8. If you have not logged in already, you will be prompted for your DCE userid and password. You must have administrator authority to create new accounts; that is, you must be part of the acct-admin user group.

If you want to set extended registry attributes, or individual account properties for this account, you must use the **dcecp account modify** command. This command is explained in more detail in the online documentation for DCE.

### Related Concepts

"SSL and Certificates" on page 277
"Certificate Authorities" on page 280
"Accounts for Component Broker Administration" on page 287

“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

“Log in a Client Principal” on page 219
“Change the Password for a Client Principal”
“Administer Accounts for Client and Server Principals” on page 316
“Chapter 11. Administer Security in your Enterprise” on page 261

## Change the Password for a Client Principal

Use this procedure to change the password for a client principal that is to use Component Broker.

**Prerequisite:** You must have created a DCE account for the client principal, as described in “Create an Account for a Client Principal” on page 317

A client principal can change its DCE password by using the rgy_edit command line utility to complete the following steps:

1. At a command line prompt, type **rgy_edit**
2. At the rgy_edit prompt, type the following command&;58;

   ```
   change -p <security_name>
   -g <group_name> -o <organization_name> -pw <new_password>
   -mp <old_password>
   ```

   Where:

   **<security_name>**
   is the security name of the principal you are changing

   **<group_name>**
   is the principal’s primary group

   **<orgnization_name>**
   is the principal’s primary organization; often this is ″none″

   **<new_password>**
   is the principal’s new password

   **<old_password>**
   is the principal’s old password.

   You must specify all three identifiers; the principal’s security name, their primary group, and their primary organization. If you leave any of these out, then you will get an error message indicating the password can use ’wildcard’ characters. If you do not know the principal’s primary group and organization, then you can enter the following command, at the rgy_edit prompt:

   ```
   view <security name>
   ```

   This command returns information about the DCE account, with the principal’s primary group and organization listed in square brackets immediately following their principal name.
3. To finish with rgy_edit, type **exit**

**Related Concepts**

“SSL and Certificates” on page 277
“Certificate Authorities” on page 280
“Accounts for Component Broker Administration” on page 287
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

“Create an Account for a Client Principal” on page 317
“Log in a Client Principal” on page 219
“Change the Password for a Server Principal”
“Administer Accounts for Client and Server Principals” on page 316
“Configure Security for a Client Style” on page 326
“Chapter 11. Administer Security in your Enterprise” on page 261

## Change the Password for a Server Principal

Use this procedure to change the password for a server principal that is to use Component Broker. To change the password for a server principal, you must change the server’s keytab file entry.

**To change the password for a server principal (in a keytab file), use the rgy_edit command line utility to complete the following steps:**

1. At a command line prompt on the host where the server resides, type **rgy_edit**
2. At the rgy_edit prompt, type the following command:

   ```
   ktadd -p <security_name> -pw
   <new_password> -r -f <keytab_filename>
   ```

   Where:

   **<security_name>**
   is the name of the server principal

   **<new_password>**
   is the new password for the server

   **-r** tells rgy_edit to synchronize the new password with the user registry

   **<keytab_filename>**
   is the fully qualified name of the keytab file on the host were this server runs.

   You must execute this command on the host where the server resides, and where the server’s keytab file is located. You must have whatever authority you have established for protecting the keytab file in the secure file system on that host, as described in “Protect Server Keytab Files” on page 315.

**Related Concepts**

“SSL and Certificates” on page 277
“Certificate Authorities” on page 280
“Accounts for Component Broker Administration” on page 287
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

"Create a Unique Keytab File for an Application Server" on page 313
"Protect Server Keytab Files" on page 315
"Change the Password for a Client Principal" on page 319
"Administer Accounts for Client and Server Principals" on page 316
"Configure Security for a Server" on page 323
"Chapter 11. Administer Security in your Enterprise" on page 261

## Enable Security within a Configuration

This overview procedure describes how you can turn on security and set the level and type of security that is appropriate for your enterprise. Details about configuring security on specific classes of objects are given in procedures listed at the end of this overview procedure.

By default, security is turned off in Component Broker system management Configurations.

To enable security within your Configuration you change the security attributes on Server Groups, Servers (freestanding) and Client Styles in that Configuration. You also have to change the security attributes for host Daemons used by that Configuration. Each class of object has the following types of security attributes:

- A set of basic security attributes that control the type of security that is enabled for those objects and the behavior of their security functions.
- A set of security attributes that control the quality of protection (QOP) that applies to communication with those objects. Because there is a large number of permutations for the quality of protection, the most common preferences are provided as a small number of models, referred to as *QOP models*. Those common preferences are indicated by the alternative values, **authenticity**, **integrity**, and **confidentiality**, of QOP-Model attributes.

  The actual quality of protection that is performed is determined by examining the QOP exported by the target server and the QOP to be performed by a given client (or server acting as a client to another server). To understand what quality of protection is actually performed, you can examine the rules specified for the perform-QOP attributes of the Client Styles and Servers. These rules are given in the procedures listed at the end of this overview procedure.

To configure the security characteristics for an object, you first set the basic security attributes as required. These attributes then filter the effect of the QOP model attributes.

To set or change any of the security attributes for a server or client style, you use the following general sequence of steps:

1. Expand your Management Zone, and the Configurations folder within that zone.
2. Expand the Configuration within which you want to set or change security.
3. Expand the folder for the class of object that you want to act on; for example, expand the Client Styles folder to act on a Client Style.
4. On the pop-up menu of the object that you want to act on, click **Edit**, to display the Object Editor window.
5. In the Object Editor window, click the security notebook tab and scroll to the attribute that you want to change.

6. Change appropriate security attributes, as described in the following detailed procedures.
7. When you have finished changing the security attributes for that object, click the **OK** button to apply the changes and close the Object Editor window.

You must repeat these steps for every Server Group, Server (freestanding) and Client Style that you want to be secure. In addition, for each host used by your Configuration, you must enable security on the host Daemon for secure communications with clients, and on the host name server if you want secure access to that server and the resources it houses. For each host Daemon that you enable to be secure, you must also set the **SSL Port** attribute on its related Protocol model.

When you have completed the security changes for all objects in your Configuration, you should activate that Configuration to apply those changes into your running enterprise.

Details about configuring security on specific classes of objects are given in the following procedures:

- "Configure Security for a Server" on page 323
- "Configure Security for a Client Style" on page 326
- "Configure Security for a Host Daemon" on page 329

**Notes:**

- Component Broker automatically creates a Name Server on every host that you configure. The Name Server is used by Component Broker to house a number of system objects, including naming contexts used in the system name space, factory finders, location objects, and so on. If you want to prevent unauthorized access to the Name Server and the resources it houses, you should "Configure Security for a Server" on page 323, as you would for your own application servers. If you will be accessing the Name Server from Java clients over the SSL-based authentication facility, you must provide a certificate for this server and include its trust-basis in the client keyrings for any clients that will access it.
- To support SSL-based authentication, you must ensure that the server's certificate and keyring are in place before you begin communicating securely with the server. This requires that you do one of the following:
  – Plan ahead to ensure the certificate you will need for any application server or name server has been prepared and signed by your designated Certificate Authority well in advance of when you will create and use the server
  – Begin by configuring the server to be insecure, and then later go back and modify the server attributes to be secure after you have obtained and prepared its certificate.

### Related Concepts

"Securing your Enterprise" on page 36
"Security" on page 263

### Related Tasks

"Chapter 11. Administer Security in your Enterprise" on page 261

## Configure Security for a Server

Use this procedure to configure the security characteristics of a server group, freestanding server, or name server. You do this by setting the security attributes of one of the following objects in your system management Configuration:

- A Server Group model, for all servers in the server group
- A Server (freestanding) model, for a freestanding server
- A Name Server model, for a name server

**Prerequisites:**

To support SSL-based authentication, you must ensure that the server's certificate and keyring are in place before you begin communicating securely with the server. This requires that you do one of the following:

- Plan ahead to ensure the certificate you will need for any application server or name server has been prepared and signed by your designated Certificate Authority well in advance of when you will create and use the server
- If needed, you can begin by configuring a server to be insecure and later go back and modify the server attributes to be secure after you have obtained and prepared its certificate.
- The name server must be secure before any application server is made secure. More generally, all servers must be secure is any server is made secure.
- Optionally, the Host Daemon should be secured to prevent intermediate attacks on the server binding protocol. However, this is optional and can be avoided if the additional overhead implied in doing so exceeds the benefit that it brings.

**To configure the security characteristics of a server group or a freestanding server, complete the following steps:**

1. Expand your Management Zone, and the Configurations folder within that zone.
2. Expand the Configuration within which you want to set or change security.
3. Expand the Server Groups folder or Servers (freestanding) folder.
4. On the pop-up menu of the Server Group or Server (freestanding) that you want to act on, click **Edit**, to display the Object Editor window.
5. In the Object Editor window, click the **Security Service** tab to display the notebook page of standard security attributes. Set these attributes to specify the type and behavior of security that you want.

   **security enabled**

   Set this attribute to **yes** to enable security in the server, to apply security to either in-bound or out-bound messages, and form credentials in the server. by default, this attribute is **no**.

   **login source**

   Set this attribute, to indicate from where the server should get its default principal login information, to one of the following values:

   **prompt**

   The server should display a login dialog to prompt the end-user for the login information. However, realize that most server hosts are locked in a closet and unattended. If this option is set, an end-user must log in the server every time the server is started.

**key table**

The server should get the login information from the keytab file specified on the **Key-tab File Name** attribute.

**environment**

The server should get the login information from the userid and password environment variables.

by default, this attribute is **key table**.

**login timeout**

Set this attribute to the number of seconds that the login dialog prompt should be displayed if no one enters the login information. If the **Login Source** attribute is set to **prompt**, or if for any other reason the server presents a login prompt, and no one enters the login information, then the server will remove the login prompt after the **Login Timeout** period, and proceed to operate as best as it can without authenticating the server's principal. In many cases, this limits what the server can do, because most significant work can be done only under an authenticated credential. by default, this attribute is **60** seconds.

**keytab file name**

Set this attribute to the fully-qualified path and file name of the keytab file used by this server. On the Server Group model, Server (freestanding) model, or Name Server model this attribute defaults to the actual path where Component Broker is installed. In the Image, this attribute should be set to the location where the keytab file was actually created when the server is created and an account established for it in the DCE user registry. By default, this attribute is WIN **c:\Cbroker\data\v5cbctab** or AIX **/usr/lpp/Cbroker/data/v5cbctab**

**delegate credentials**

This attribute indicates whether the credentials of the requesting principal should be delegated on downstream requests. *This attribute is ignored in this release.*

**DCE client association enabled**

This attribute indicates whether clients of this server are allowed to create secure associations based on DCE. This determines whether a DCE-tagged component is included in exported IORs. *This attribute can only be set to* **yes** *in this release.*

**SSL Type-I client association enabled**

Set this attribute to **yes** to specify that clients of this server are allowed to create secure associations based on SSL using the Type-I authentication model. This means that an SSL-tagged component is included in exported IORs. by default, this attribute is **yes**.

**SSL keyring file**

Set this attribute to the fully-qualified path and file name of the keyring containing the server's certificate and private key. by default, this attribute is set to WIN **c:\Cbroker\data\keyrings\<server-name>.kdb**

and AIX **/usr/lpp/Cbroker/data/keyrings/<server-name>.kdb**

where <server-name> is the security name for this server, which is set automatically on the Server Image. For a server that is a member of a server group, if you want to change the value of the SSL keyring file attribute, you can do so by editing the Server Image. For freestanding servers and name servers, this attribute can be changed on the Server model.

**SSL keyring password**
This attribute specifies the password used by IKMGUI to store the server's private key in the keyring file. This password is not used to protect the file and so the password itself does not have to be protected. It is only required to release the information stored by IKMGUI during runtime. by default, this attribute is **Cbroker**. If you change this attribute you must be certain that the password matches the one you actually defined for the keyring file in IKMGUI.

**SSL V3 session timeout**
Set this attribute to the lifetime, in the range 0 through 86400 seconds (1 day), of any connections formed with clients of the server. Connections are automatically recreated when their session information expires, and so setting this to anything less than 86400 (1 day) is only needed if the sensitivity of the information in the server warrants creating new session keys more frequently. by default, this attribute is 86400 seconds (1 day). If you set this attribute to 0, then sessions will never timeout and consequently sessions are not renegotiated.

**standard export QOP models**
Set this attribute to the standard type of QOP model that will be exported to any clients of this server. The QOP model specified by this attribute determines the QOP that clients of this server must (*qop ... required*) and can (*qop ... supported*) perform in any communication with the server. You can set this attribute to one of the following values: **authenticity**, **integrity**, and **confidentiality**. by default, this attribute is **authenticity**.

The effect of this attribute is given in "Standard Export QOP Models for Servers" on page 455.

**standard perform QOP models**
Set this attribute to the standard type of QOP model that will be performed by this server when communicating with any other downstream server. That is, the QOP specified by this attribute will determine what QOP this server will use given the QOP choices exported by the server it is communicating with. You can set this attribute to one of the following values: **authenticity**, **integrity**, and **confidentiality**. by default, this attribute is **authenticity**.

The effect of this attribute is given in "Standard Perform QOP Models for Servers" on page 458.

6. When you have finished changing the security attributes for the server, click the **OK** button to apply the changes and close the Object Editor window.

When you have completed the security changes for all objects in your Configuration, you should activate that Configuration to apply those changes into your running enterprise. This also updates the attributes of the Server Image, including setting the **security name** attribute for the server. A server's security name is set automatically, by first concatenating the name of the server host and the server name separated by a dash, then confirming its uniqueness in the DCE user registry. If the server's security name is not unique, then the server name is modified by suffixing it with the monotonically increase numeric string until a unique name is found.

**Note:** The actual quality of protection that is performed is determined by the QOP *exported* by the target server and the QOP to be *performed* by a given client (or server acting as a client to another server). To understand what quality of protection is actually performed, you can examine the Perform QOP Models attributes of the Client Styles and Servers:

- “Standard Perform QOP Models for Client Styles” on page 460
- “Standard Perform QOP Models for Servers” on page 458

### Related Concepts

“SSL and Certificates” on page 277
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

### Related Tasks

“Configure Security for a Client Style”
“Configure Security for a Host Daemon” on page 329
“Enable Security within a Configuration” on page 321
“Create and Install Server Certificates” on page 292

### Related References

“Standard Export QOP Models for Servers” on page 455
“Standard Perform QOP Models for Servers” on page 458
“Standard Perform QOP Models for Client Styles” on page 460

## Configure Security for a Client Style

Use this procedure to configure the security characteristics of a client style. You do this by setting security attributes of the Client Style model, which are applied to all clients that use that model.

**Notes:**

- If you install the DCE client at a Java client you have the choice of using either DCE or SSL-based authentication. You control which is used through security attributes in the Client Style model for that Java client. If you do set both DCE and SSL enabling attributes to **yes**, DCE takes precedence and DCE-based security is used from the client.
- If several clients need different security policies, you must create a different client style for each combination of security policies that you want to use.

**To configure the security characteristics of a client style, complete the following steps:**

1. Expand your Management Zone, and the Configurations folder within that zone.
2. Expand the Configuration within which you want to set or change security.
3. Expand the Client Styles folder folder.
4. On the pop-up menu of the Client Style that you want to act on, click **Edit**, to display the Object Editor window.
5. In the Object Editor window, click the **Security Service** tab to display the notebook page of standard security attributes. Set these attributes to specify the type and behavior of security that you want.

**Security Enabled**

Set this attribute to **yes** to enable security in the client, to apply security to out-bound messages, and form credentials in the client. By default, this attribute is **no**.

**Login Source**

Set this attribute, to indicate from where the client should get its default principal login information, to one of the following values:

**prompt**

The client should display a login dialog to prompt the end-user for the login information. If this option is set, an end-user must log in the client every time the client is started.

**environment**

The client should get the login information from the userid and password environment variables.

By default, this attribute is **prompt**.

This attribute does not apply to Java clients that use SSL-based security.

**login timeout**

Set this attribute to the number of seconds that the login dialog prompt should be displayed if no one enters the login information. If the **Login Source** attribute is set to **prompt**, or if for any other reason the client presents a login prompt, and no one enters the login information, then the client will remove the login prompt after the **Login Timeout** period, and proceed to operate as best as it can without authenticating the client's principal. In many cases, this limits what the client can do, because most significant work can be done only under an authenticated credential. By default, this attribute is **60** seconds.

**DCE server association enabled**

Set this attribute to enable the client to create secure associations with a server based on DCE. This determines whether a DCE-tagged component is included in exported IORs. By default, this attribute is **no**.

A Java client cannot use both DCE and SSL-based authentication. If both the **DCE Server Association Enabled** and **SSL Type-I Server Association Enabled** attributes are set to **yes**, then **DCE Server Association Enabled** takes precedence and DCE-based security will be used from the Java client.

**SSL Type-I server association enabled**

Set this attribute to **yes** to enable the client to create secure associations with a server based on SSL using the Type-I authentication model. By default, this attribute is **yes**.

**SSL keyring file**

Set this attribute to the fully-qualified Java class name of the keyring containing the certificates for the trust-basis that should be recognized by this client. (This is generally a class on the client's Web Server.) By default, this attribute is set to the name of the client style; we assume you will use KEYMAN to set the class name of the client keyring to match the name of the client style that will refer to it. However, if you use some other class name, you must set this attribute to match that class name.

Make sure that this class can be accessed in the CLASSPATH or CODEBASE of the Java client.

**SSL keyring password**
This attribute specifies the password used by KEYMAN to store the trust basis fpr the client in the keyring file. This password is not used to protect the file and so the password itself does not have to be protected. It is only required to release the information stored by KEYMAN during runtime. By default, this attribute is **CBroker**. If you change this attribute you must be certain that the password matches the one you actually defined for the keyring file in KEYMAN.

**SSL V3 Session Timeout**
Set this attribute to the lifetime, in the range 0 through 86400 seconds, of any SSL-based connections that the client forms with servers. Connections are automatically recreated when their session information expires, and so setting this to anything less than 86400 (1 day) is only needed if the sensitivity of the information in the client warrants creating new session keys more frequently. By default, this attribute is 86400 seconds (1 day). You cannot set the value greater than 86400. If you set this attribute to 0, then sessions will never timeout and consequently sessions are not be renegotiated.

**SSL Credentials Timeout**
This attribute is not currently used.

**Standard Perform QOP Models**
Set this attribute to the type of QOP model that will be performed by this client when communicating with a server. That is, the QOP specified by this attribute will determine what QOP this client will use given the QOP choices exported by the server it is communicating with. You can set this attribute to one of the following values: **authenticity**, **integrity**, and **confidentiality**. By default, this attribute is **authenticity**.

The effect of this attribute is given in “Standard Perform QOP Models for Client Styles” on page 460.

6. When you have finished changing the security attributes for the client style, click the **OK** button to apply the changes and close the Object Editor window.

When you have completed the security changes for all objects in your Configuration, you should activate that Configuration to apply those changes into your running enterprise.

### Related Concepts

“SSL and Certificates” on page 277
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

### Related Tasks

“Configure Security for a Server” on page 323
“Configure Security for a Host Daemon” on page 329
“Enable Security within a Configuration” on page 321
“Add a Server Certificate to a Client Keyring Class” on page 310.

**Related References**

"Standard Export QOP Models for Servers" on page 455
"Standard Export QOP Models for Host Daemons" on page 457
"Standard Perform QOP Models for Servers" on page 458
"Standard Perform QOP Models for Client Styles" on page 460

## Configure Security for a Host Daemon

Use this procedure to configure the security characteristics of a host daemon. You do this by setting security attributes of the Daemon Image, under the Host Image for the host on which the daemon runs. For each host daemon that you enable to be secure, you must also set the **SSL Port** attribute on the host's Protocol model. This procedure is therefore split into the following two stages, which can be completed in any order:

1. Set the security attributes of the Daemon Image (page 329)
2. Set the SSL port attribute of the Daemon's protocol

**Note:** You must complete this procedure for every host on which you want the daemon to use secure SSL-based communication with clients.

**To configure the security attributes of a host Daemon Image, complete the following steps:**

1. Start the System Manager user interface and (if needed) set the user-level to **Expert**.

   This displays the Host Images folder on the home view of the system management network.
2. Expand the Host Images folder, to display the Image for the host on which the daemon runs.
3. Expand the Host Image, to display the Daemon Images folder.
4. Expand the Daemon Images folder.
5. From the pop-up menu of the Daemon Image, click **Edit** to display the Object Editor window.
6. In the Object Editor window, click the **Security Service** tab to display the notebook page of standard security attributes. Set these attributes to specify the type and behavior of security that you want.

   **security enabled**
   Set this attribute to **yes** to enable security in the daemon, to apply security to in-bound messages. By default, this attribute is **no**.

   **security name**
   This attribute is set automatically to the security name for the daemon, by concatenating the name of the host and the string **Daemon** separated by a dash.

   **SSL Type-I client association enabled**
   Set this attribute to **yes** to specify that clients of this daemon are allowed to create secure associations based on SSL using the Type-I authentication model. This means that an SSL-tagged component is included in exported IORs. By default, this attribute is **yes**.

   **SSL keyring file**
   Set this attribute to the fully-qualified path and file name of the keyring containing the daemon's certificate and private key. By default, this

attribute is set to WIN **c:\CBroker\data\daemon.kdb**″ and AIX **/usr/lpp/CBroker/data/daemon.kdb**

**SSL keyring password**

This attribute specifies the password used by IKMGUI to store the daemon’s private key in its keyring file. This password is not used to protect the file and so the password itself does not have to be protected. It is only required to release the information stored by IKMGUI during runtime. By default, this attribute is **CBroker**. If you change this attribute you must be certain that the password matches the one you actually defined for the keyring file in IKMGUI.

**SSL V3 session timeout**

Set this attribute to the lifetime, in seconds, of any connections formed with clients of the daemon. Connections are automatically recreated when their session information expires, and so setting this to anything less than 86400 (1 day) is only needed if the sensitivity of the information in the daemon warrants creating new session keys more frequently. By default, this attribute is 86400 seconds (1 day). You cannot set the value greater than 86400. If you set this attribute to 0, then sessions will never timeout and consequently sessions are not be renegotiated.

**standard export QOP models**

Set this attribute to the type of QOP model that will be exported to any clients of this daemon. The QOP model specified by this attribute determines the QOP that clients of this daemon must (*qop ... Required*) and can (*qop ... Supported*) perform in any communication with the daemon. You can set this attribute to one of the following values: **authenticity**, **integrity**, and **confidentiality**. By default, this attribute is **authenticity**.

The effect of this attribute is given in “Standard Export QOP Models for Host Daemons” on page 457.

7. When you have finished changing the security attributes for the host daemon, click the **OK** button to apply the changes and close the Object Editor window.
8. You should also set the number of the port used by clients for secure SSL-based communication with the daemon.

**To configure the SSL port for a host daemon, complete the following steps:**

1. Expand the Hosts folder, to display the model for the host on which the daemon runs.
2. Expand the Host model, to display the Configured TCP/IP Protocol relationship folder.

   If this folder is not displayed, check that the filter is set to show relationships.
3. Expand the the Configured TCP/IP Protocol relationship folder.
4. From the pop-up menu of the TCP/IP Protocol, click **Edit** to display the Object Editor window.
5. In the Object Editor window, click the **Security Service** tab to display the notebook page of standard security attributes. Set the **SSL Port** attribute to the number of the port that should be used by clients to form a secure association with the host daemon based on SSL. This port is used with SSL-based authentication for communication with the host daemon. By default, this attribute is 3004.

6. When you have finished changing the security attributes for the protocol, click the **OK** button to apply the changes and close the Object Editor window.

When you have completed the security changes for all objects in your Configuration, you should activate that Configuration to apply those changes into your running enterprise.

**Related Concepts**

“SSL and Certificates” on page 277
“Models of Trust Validation” on page 282
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

“Configure Security for a Server” on page 323
“Configure Security for a Client Style” on page 326
“Enable Security within a Configuration” on page 321
“Add a Server Certificate to a Client Keyring Class” on page 310.

**Related References**

“Standard Export QOP Models for Host Daemons” on page 457
“Standard Perform QOP Models for Client Styles” on page 460

## Disable the Security Service

Use this procedure to disable the Component Broker Security Service. This involves disabling the Security Service for servers and for client styles.

Normally, you disable the Security Service by changing attributes of objects defined in your system management Configuration, as described in this procedure. After making these changes, you must activate the Configuration again, to apply these (and any other changes) consistently across all the servers and client styles changed in the Configuration.

(You can make the same changes to Server Images and Client Style Images to disable the Security Service directly on real servers and clients, but such changes would be lost when you next activate your system management Configuration.)

**To disable the Security Service for a server, complete the following steps:**

1. Display the Server Group model or Server (freestanding) model that defines the attributes of the server.
2. From the pop-up menu for the Server Group model or Server (freestanding) model, click **Edit**. This displays the Object Editor notebook for the model.
3. Click the **Security Service** tab.
4. Change the **security enabled** attribute from **yes** to **no**.
5. To apply the changes and close the Object Editor, click the **OK** button.

**To disable the Security Service for a client style, complete the following steps:**

1. Display the Client Style model that defines the attributes of the clients.

2. From the pop-up menu for the Client Style model, click **Edit**. This displays the Object Editor notebook for the model.
3. Click the **Security Service** tab.
4. Change the **security enabled** attribute from **yes** to **no**.
5. To apply the changes and close the Object Editor, click the **OK** button.

When the Configuration is next activated, the client style properties file is updated with security disabled.

To disable the Security Service for Java clients, you need to make the updated client style properties available to the Java clients, then specify the URL of the updated properties file when a Java client is next used. For example, to run PolicyApp with security disabled, you could enter the following command:

```
java -Dcom.ibm.CORBA.ClientStyleImageURL=url_of_properties_file
     -Dcom.ibm.CORBA.BootstrapHost=bootstrap_host_name
     -Dcom.ibm.CORBA.BootstrapPort=port PolicyApp
```

**Where:**

**-Dcom.ibm.CORBA.ClientStyleImageURL=***url_of_properties_file*
is used to specify the URL of the client style properties file to be used by this client. That properties file has had security disabled.

**-Dcom.ibm.CORBA.BootstrapHost=***bootstrap_host_name*
is used to specify the bootstrap host name that should be used by this client.

**-Dcom.ibm.CORBA.BootstrapPort=***port*
is used to specify the bootstrap host port number that should be used by this client.

### Related Concepts

“Security in a Component Broker Network” on page 263
“Security” on page 263
“Use of DCE- and SSL-Based Security in Your Enterprise” on page 271
“Chapter 8. Administer Clients” on page 205

### Related Tasks

“Configure Security for a Server” on page 323
“Configure a Server for Connections to tier-3 Systems” on page 392
“Configure Security for a Client Style” on page 326
“Configure Security for a Host Daemon” on page 329
“Enable Security within a Configuration” on page 321

# Chapter 12. Administer Workload Management

To setup workload management for a server group, you configure it as a **controlled server group**. A controlled server group is a server group that has been configured with a *managing host*. The managing host provides the management services that control workload distribution across the servers in the group. For more information, see “Configure a Controlled Server Group” on page 339.

When you activate a Configuration that contains a controlled server group, the Activate action starts the application servers that are members of the server group, but also builds and starts special servers used for the server group. A *Server Group Control Point (SGCP)* server is built to control the server group. A *Server Group Gateway (SGGW)* server is built so that CBConnector can present the server group as a single server to non WLM-enhanced clients.

If needed later, you can remove a complete controlled server group from your enterprise. This removes the application servers that are members of the server group from their hosts then removes the group’s SGCP and SGGW servers from their hosts. This also removes the servers’ entries from enterprise services such as DCE and the Host Naming Service. For more information, see “Remove a Controlled Server Group from Your Enterprise” on page 341.

For more information about Component Broker workload management, see “Workload Management” on page 335.

### An Overview of a Workload Management Configuration

This topic describes the basic configuration that you use for workload management.

**Server Group**

The Server Group model defines the *controlled server group* for workload management. The controlled server group can process work requests for applications represented by Application models configured on the Server Group. CBConnector determines the best target Server (member of group) model for the work request based on the state of each server, as described in “Workload Management” on page 335.

The **Configure Managed Server Group** action is used to configure the Server Group model onto one Host. That host becomes the **Managing Host** on which the SGCP server and SGGW server run for the controlled server group.

**Policy Group**

The Application Family contains a Policy Group that defines how *workload* **managed** objects in a container are managed. The policy group is not normally administered, so does not appear in the system management Configuration.

**Container**

The Application Family contains a Container used to store *workload* **manageable** objects. The container for workload manageable objects is not normally administered, so does not appear in the system management Configuration.

Any application may be installed on a controlled server group. However only applications which include designated workload managed objects will benefit from automatic workload distribution.

When you activate a workload management Configuration, the System Manager makes your enterprise match the definitions within that Configuration, like the enterprise shown in Figure 64.

**Notes on the figure:**

1. The Server Group model was configured onto the Host model for Host 1, as the managing host. When the Configuration was activated, the SGCP server and SGGW server were created automatically on that host.
2. The application servers, SA1, SA2, SA3, and SA4 are the target servers for workload distribution.

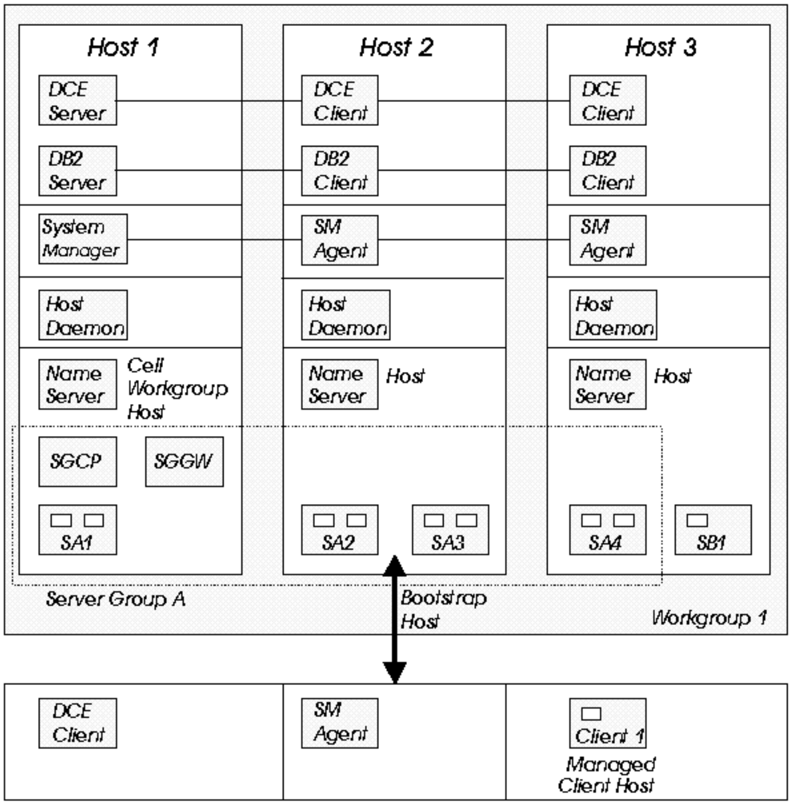


*Figure 64. An Enterprise Including a Controlled Server Group*

### Related Concepts

"Workload Management" on page 335
"Controlled Server Groups" on page 336
"Server Group Control Point (SGCP)" on page 337
"Server Group Gateway (SGGW)" on page 338
"Workload Management (WLM)-Enhanced clients" on page 338

### Related Tasks

"Configure a Controlled Server Group" on page 339

## Workload Management

*Workload Management*, is the discipline of defining, monitoring, and actively managing work in a system network. In a Component Broker context *work* is taken to mean the dispatch, routing, and receipt of requests between objects in the distributed network and their eventual execution within an application server. Component Broker has a number of features which enable workload management:

- Workload distribution (page 335)
- “Controlled Server Groups” on page 336
- “Monitoring the Health of Application Servers” on page 7
- “Centralized Configuration of System Management Objects” on page 7

### Workload Distribution

A client application generates work by making requests to one or more objects managed by an application server. If you use a server group to define multiple application servers then you increase the capacity of your enterprise to process work on behalf of an application. However, if clients are in some way configured to always use a fixed server, there is still the potential for one server to be heavily loaded while another is idle. Clients need to be able to distribute their work between the available servers. Component Broker provides a simple *workload distribution* mechanism which exploits an enhanced server group configuration in which the server group is associated with a *managing host*. The Managing Host provides the management services which control the work distribution. Server groups which support workload distribution in this way are referred to as *controlled server groups*.

Applications must designate which managed objects can support workload distribution. Such objects are referred to as workload managed objects and have been specially coded within certain guidelines and restrictions. This special coding may not be appropriate for all objects. Workload distribution offers improved scaleable performance as it seeks to share, and ideally balance, workload across the available servers in a server group. In a heavily loaded system this should increase throughput and decrease the average response time of method invocations. Workload distribution also offers improved availability as it removes the fixed association between an object and a server, allowing use of that object providing at least one of the servers in the group is available. This potentially allows a client application to recover more quickly from server failures.

The Component Broker workload distribution mechanism provides for an application to configure policy information that determines how to select a server from those available and when workload can be switched to a different server. This information is known as the *binding policy* that determines which server to choose, and the *binding affinity* that determines how long to stick with the chosen server. This information is managed by Policy Group configuration objects.

There is currently only one binding policy and binding affinity configuration available. The default binding policy uses a random server selection algorithm which chooses a server at random from those in the controlled server group known to be available. This is a ″static″ strategy that does not base a server selection decision on any information about the system. In particular, it does not take into account the current load on servers in the group, the distribution of servers in the group across hosts or the relative computer power of each host. The default bind affinity rules establish a single server affinity between a client or server process and the selected server in each server group. The effect of this is to bind a process to a randomly chosen server and to continue to use the same server to process all requests for workload managed objects for that server group. If the chosen server becomes unavailable, for whatever reason, the client affinity is broken and a new selection may be made. Client application programs must be written to take advantage of the increased availability of services in this environment.

Requests that are routed through the controlled server group gateway (SGGW) server share the same binding affinity. The effect of this is that the first request through the SGGW server chooses which server in the controlled server group to use. That same server will be used for *all* client programs that use the SGGW server, until it becomes necessary to choose another server in the controlled server group.

**Related Concepts**

“Controlled Server Groups”
“Monitoring the Health of Application Servers” on page 7

**Related Tasks**

“Define and Configure Servers and Server Groups” on page 182
“Configure a Controlled Server Group” on page 339

## Controlled Server Groups

The main concept of Component Broker workload management is the **controlled server group**. A controlled server group is a server group that has been configured with a *managing host*. The managing host provides the management services that control workload distribution across the servers in the group.

The servers in a server group are configured to be generally alike by the fact that their configuration is determined by a single *Server Group model*. For example, the applications used by the server group are configured onto the Server Group model. Therefore, the applications installed on each server are the same. However there are some assumptions made about the host environment on the computers used by the group’s servers. Because each server in the group must be able to equally process requests from clients to specific workload managed objects, it is implicitly required that access has been configured to common (that is, shared) datastores or tier-3 systems. While each server in the group is configured with the same connection aliases, the flexibility of, for example, DB2 configuration does not automatically ensure that the same physical database is accessed. This is an administrator responsibility.

Note that a special case exists for objects providing a read-only interface. It is possible to configure each server in a group to access a private copy of a read-only persistent datastore if the nature of the application makes this beneficial.

While servers in a server group may be deployed across many hosts, a futher restriction is required for a controlled server group. The hosts on which the servers in the same server group run must all ″prefer″ the same work group. This allows the server group to appear as a single work group resource. Therefore, all the Hosts related by *Configured Hosts* relationships to the Server (Member Of Group) models must have the same work group as their *Preferred Work Group*.

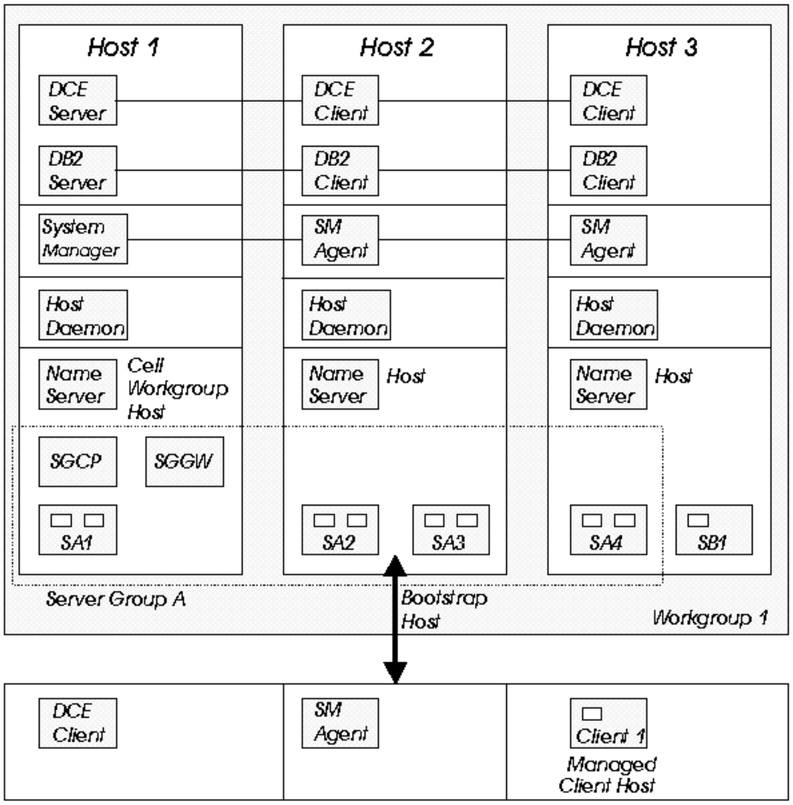


*Figure 65. A Component Broker Enterprise Containing a Controlled Server Group*

When you activate a Configuration that contains a controlled server group, the Activate action starts the application servers that are members of the server group, but also builds and starts special servers used for the server group. A *Server Group Control Point (SGCP)* server is built to control the server group. A *Server Group Gateway (SGGW)* server is built so that CBConnector can present the server group as a single server to non WLM enhanced clients.

**Related Concepts**

“Workload Management” on page 335
“Server Group Control Point (SGCP)”
“Server Group Gateway (SGGW)” on page 338
“Workload Management (WLM)-Enhanced clients” on page 338

**Related Tasks**

“Configure a Controlled Server Group” on page 339

## Server Group Control Point (SGCP)

A controlled server group is managed by a special server that is known as the **Server Group Control Point (SGCP)**. The Server Group Control Point coordinates the exchange of information between servers and clients, allowing clients to know which servers in the group are active. This configuration data is normally cached by both clients and servers. The Server Group Control Point ensures that the latest information is always used, as configurations are changed dynamically by the Systems Administrator.

The Server Group Control Point for each controlled server group is started automatically.

**Related Concepts**

“Controlled Server Groups” on page 336
“Server Group Gateway (SGGW)”
“Workload Management (WLM)-Enhanced clients”

**Related Tasks**

“Configure a Controlled Server Group” on page 339

### Server Group Gateway (SGGW)

A *Server Group Gateway (SGGW)* is a special server found on the Managing Host of a Controlled Server Group. The SGGW server uses the workload distribution mechanism to route requests to application servers in the group. Requests processed by the SGGW server typically originate from client programs that use an ORB which does not support client-initiated workload distribution. This currently includes client programs using the Java ORB and Microsoft Visual C++ ORB supplied with Component Broker. The SGGW server needs information about an object’s interface in order to route requests to an object instance in an appropriate server. The SGGW server relies on this information having been loaded into the Interface Repository on the Managing Host computer.

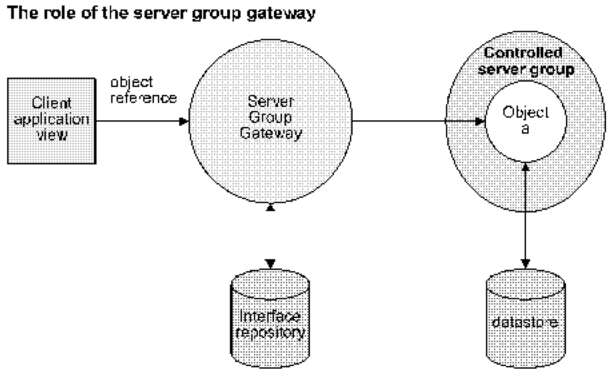


*Figure 66. Server Group Gateway (SGGW) Server*

**Related Concepts**

“Workload Management” on page 335
“Controlled Server Groups” on page 336
“Server Group Control Point (SGCP)” on page 337
“Workload Management (WLM)-Enhanced clients”

**Related Tasks**

“Configure a Controlled Server Group” on page 339

## Workload Management (WLM)-Enhanced clients

Client programs which use the Component Broker VisualAge C++ ORB benefit from an ORB extension that implements a simple Workload Distribution mechanism. These clients are referred to as being *WLM-enhanced*. When the ORB is required to dispatch a request to a remote workload managed object, a WLM-enhanced client is able to use a binding policy to select one of the available servers in the group to process the request. Clients that do not have the ORB extension must route requests to workload managed objects through the Server Group Gateway.

**Related Concepts**

“Workload Management” on page 335
“Controlled Server Groups” on page 336
“Server Group Control Point (SGCP)” on page 337
“Server Group Gateway (SGGW)” on page 338

**Related Tasks**

“Configure a Controlled Server Group”
Disable the ORB’s Enhanced Workload Management (WLM) Extension

## Configure a Controlled Server Group

To configure a controlled server group, you must deploy the servers in your network, and then locate the Server Group Control Point server and the Server Group Gateway for each controlled server group. This topic shows you how to do this.

**Note:** If you want a server group to be a controlled server group (for workload management), you must configure it before you first activate the Configuration that contains the Server Group model, as described in this topic. *You should not reconfigure a server group to be a controlled server group after the Configuration that contains the Server Group model has been activated at least once.*

Before carrying out this task, you should make sure you are familiar with:

- “Controlled Server Groups” on page 336
- “Server Group Control Point (SGCP)” on page 337
- “Server Group Gateway (SGGW)” on page 338

**To configure a controlled server group, follow these steps:**

1. Start the SM User Interface.
2. Open (or create) your chosen Management Zone
3. Within the Management Zone open (or create) your chosen Configuration.
4. From the Configuration icon’s menu, select **New - Server Group**.
5. Type the name of your new Server Group, and select **OK**.
6. From the newly created Server Group’s icon, select **Drag**.
7. In the Hosts folder, locate the required Host on which the Server Group Control Point (SGCP) server and the Server Group Gateway (SGGW) are to run. If some hosts are more reliable or available than others, you may wish to select one of those hosts to manage the controlled server group.
8. From the Host’s pop-up menu, click **Configure Managed Server Group**. This has now designated the server group as a controlled server group.

   Create and configure each server within the server group, by repeating steps 9 (page 339) through 13.
9. From the newly created Server Group’s icon, select **New - Server (member of group)**.

10. Type the name of your new Server, and select **OK**. Remember that the name of each Server must be unique across your enterprise and cannot be the same as the name of a Server Group or a Server (Freestanding).
11. From the pop-up menu of the newly created Server, select **Drag**.
12. In the Hosts folder, locate the required Host model for the host on which the server is to run.
13. From the pop-up menu of the Host, select **Configure Server (member of group)**. Servers in a controlled server group should be placed on hosts with regard to the likely network traffic and relative load on each host. One reasonable approach is to place each server on a different host. The normal considerations affecting how many servers to configure on a single host are no different in the controlled server group environment.
14. Repeat steps 9 to 13 for each server you want in the controlled server group.

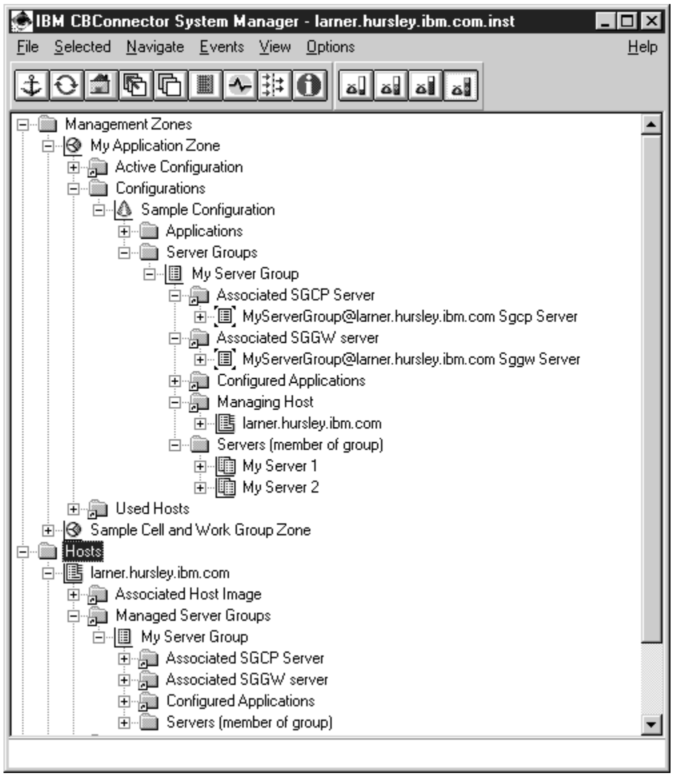


*Figure 67. The System Manager user interface, showing a configured controlled server group*

*The above snapshot is taken from the Windows NT version of Component Broker. The relationships shown are the same on the AIX version.*

You have now configured a controlled server group. This will take effect when you next activate the updated configuration.

### Related Concepts

"Controlled Server Groups" on page 336
"Server Group Gateway (SGGW)" on page 338
"Workload Management (WLM)-Enhanced clients" on page 338

**Related Tasks**

"Configure a Controlled Server Group" on page 339

## Remove a Controlled Server Group from Your Enterprise

Use this procedure to remove a complete controlled server group from your enterprise. This removes the application servers that are members of the server group from their hosts then removes the group's SGCP and SGGW servers from their hosts. This also removes the servers' entries from enterprise services such as DCE and the Host Naming Service.

If you want to remove an individual application server from its host, complete the steps described in "Remove an Application Server from a Host" on page 196.

If you want to remove a name server from a host, complete the steps described in "Remove a Name Server from a Host" on page 176.

**To remove a complete controlled server group, complete the following steps:**

1. Delete the Server Group model that defines the server group from its Configuration then reactivate the Configuration, as described in "Delete Objects from an Active Configuration" on page 259.

   This updates the active configuration and stops the server group's servers to be removed. An Action Console window is displayed where you can monitor the progress of the action. When a server is able to be removed, the console displays a message *″xxx server has been deactivated and can now be deleted″*. Note that before this message is displayed, servers being deactivated may be started then stopped one or more times if needed.

   Wait until the activate action has completed, as indicated by the **completed** message in the Action Console window, before continuing with the next step.

2. Check that all the applications configured on the server group have been removed from its application servers, to ensure that their registered objects (such as homes) have been removed from the name tree. To ensure that this process was successful, complete the follow checks:

3. 
   - For each the Server Image, check the Application Images to see if any of your configured applications are still there after the process has completed.
     - For application servers, iObjectServices should be the only application.
     - For the SGCP's Server Image, with a name like *wlmsrvgrp@myhost Sgcp Server*, iSgcpServices should be the only application.
     - For the SGGW's Server Image, with a name like *wlmsrvgrp@myhost Sggw Server*, iSggwServices should be the only application.
   - Using the DCE Director, go to the local root for each application server's host and check the path host/resources/servers/<servername>/collections and see if there are any Homes bound into this context that are from your applications (note that there will be some still there from the iObjectServices application)
   - Using the DCE Director, go to the local root for each application server's host and check the path host/resources/factories/somlcRepository/serverBranch/<servername>/collections and see if there are any Homes bound into this context that are from your applications (note that there will be some still there from the iObjectServices application)

If the above checks show your applications have been deleted from the server, it is now safe to remove the Server Image for each application server, as described in step 3 (page 342).

**Note:** Do not remove the SGCP's Server Image and SGGW's Server Image at this time; they will be removed later.

4. Remove each application server in the server group, by repeating steps 4 (page 342) to 5 for each server.

   **Note:** Instead of repeating steps 4 and 5 for each server, you can remove several servers at the same time. To do this, switch to List view, select their Server Images, then click **Selected - Remove** from the menu bar.
5. Display the Server Image for each application server in the server group. (To do this, you may need to "Control Which Objects are Displayed" on page 62 to at least **Expert**.)
6. On the pop-up menu of the Server Image, click **Remove**.

   This starts an asynchronous action to remove the server from its host.

   Besides checking the changes made by this task, you can use the DCE Director to perform the following checks:

   - Check that the server name (<servername>) is not bound into the context host/resources/servers and the context host/resources/factories/somlcRepository/serverBranch.
   - When you have removed all application servers, check that the server group name is not bound into the context work group/resources/factories/somlcRepository/serverBranch.
7. Remove the server group control point and gateway, by completing steps 7 (page 342) to 10.

   **Note:** Wait until the actions to remove application servers have completed before starting the following steps.
8. Display the Server Image for the server group's SGCP server. This has a name like *wlmsrvgrp@myhost Sgcp Server* and, like any other Server Image, is found under **Host Images - myhost - Server Images**.
9. On the pop-up menu of the SGCP's Server Image, click **Remove**.

   This starts an asynchronous action to remove the server from its host. Wait until the action has completed before starting step 9 (page 342).
10. Display the Server Image for the server group's SGGW server. This has a name like *wlmsrvgrp@myhost Sggw Server* and, like any other Server Image, is found under **Host Images - myhost - Server Images**.
11. On the pop-up menu of the SGGW's Server Image, click **Remove**.

### Related Tasks

"Change the Active Configuration of Your Enterprise" on page 431
"Remove an Application Server from a Host" on page 196
"Remove a Name Server from a Host" on page 176
"Uninstall Managed Hosts" on page 178
"Activate a Configuration" on page 256
"Delete Objects from an Active Configuration" on page 259
"Control Which Objects are Displayed" on page 62
"Act on Several Selected Objects" on page 68

# Chapter 13. Administer Connections to Tier-3 Systems

This topic provides information about administering connections between Component Broker application servers and tier-3 systems. For an introduction to the types of connections supported by Component Broker, see “Connections to Tier-3 Systems” on page 29.

The tasks that you need to complete depend on the type of connection that you want to administer.

Administration of connections to tier-3 systems involves the following general tasks:

- Configure a new connection to a tier-3 system

  Use this general task to configure the characteristics of a new connection to a tier-3 system, such that the connection can be used by one or more Component Broker application servers. For APPC and ECI connections this also involves configuring the communications product used.

  For more information, see the following task descriptions:

  – “Configure a new HOD Connection to a Tier-3 System” on page 344

  – “Configure a new ECI Connection to a Tier-3 System” on page 348

  – “Configure a new APPC Connection to a Tier-3 System” on page 353

- “Configure a Server for Connections to tier-3 Systems” on page 392

  Use this general task to configure server-wide attributes for communication between an application server and tier-3 systems. This defines the userid and password that an application server uses to connect securely to *ant* tier-3 system. This also defines the maximum number and timeouts for connections to tier-3 systems.

- Change the attributes of an existing tier-3 connection

  Use this general task to change the characteristics of a connection to a tier-3 system. The characteristics that you can change depend on the type of connection.

  For more information, see the following task descriptions:

  – “Configure a HOD Connection for an Application” on page 345

  – “Configure an ECI Connection for an Application” on page 349

  – “Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

  – “Configure a Server for Connections to tier-3 Systems” on page 392

**Related Concepts**

“HOD Connections to Tier-3 Systems” on page 32
“ECI Connections to Tier-3 Systems” on page 31
“APPC Connections to Tier-3 Systems” on page 33

**Related Tasks**

“Collect information for your SNA configuration” on page 358.
“Configure Communications Server” on page 367

## Configure a new HOD Connection to a Tier-3 System

Use this procedure to configure a connection to a tier-3 system (CICS region or IMS server) to use 3270 terminal emulation. This connection can be used by one or more applications running on Component Broker application servers.

If you want to configure another type of connection to a tier-3 system, see the related tasks at the end of this topic.

HOD communication between an application server and a tier-3 system is via Host On-Demand (HOD) and a telnet server (daemon), as shown in the following figure:

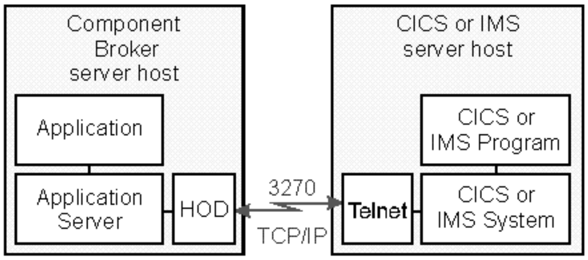


*Figure 68. HOD Connection to a Tier-3 System*

Component Broker ships a subset of Host On-Demand, which provides a Java-based TN3270 client, as part of the Procedural Applcation Adaptor (PAA). Host On-Demand is configured automatically when you add PAA services to an application server.

For information about configuring the telnet server (3270 communications) for a tier-3 system, see the information provided with the tier-3 system. For example, see the *CICS Intercommunication Guide*.

You should record your values for the following parameters to be used later when configuring the HOD Connection:

*Table 17.* **Parameters used to Configure a HOD Connection to a Tier-3 System**

| Parameter | Example Value | Notes |
|---|---|---|
| **hostname** | my.cics.host | The TCP/IP hostname of the tier-3 system on which the telnet listener is running. From the tier-3 system configuration. |
| **port number** | 23 | The port number used by the telnet listener. From the tier-3 system configuration. |

To configure a HOD connection with a tier-3 system, use the System Manager user interface to complete the following tasks:

1. “Configure a HOD Connection for an Application” on page 345.
   Complete this task to specify connection details by editing a HOD Connection model, which is normally created automatically by the application that need to use the connection.
2. “Configure the iPAAServices application onto the application server” on page 394

Complete this task only if you need to add the Procedural application adaptor (PAA) services (iPAAServices application) to an application server. These services enable an application server to communicate with tier-3 systems. If the application server already has the iPAAServices application (perhaps added for another tier-3 connection), you do not need to complete this task again.

3. “Configure a Server for Connections to tier-3 Systems” on page 392
   Complete this task only if you need to define the userid and password that an application server uses to connect securely with *any* tier-3 system or if you need to change server-wide connection limits. If you have already specified the userid, password, or server-wide limits, you do not need to complete this task again.

**Related Concepts**

“HOD Connections to Tier-3 Systems” on page 32
“Connections to Tier-3 Systems” on page 29

**Related Tasks**

“Configure a new ECI Connection to a Tier-3 System” on page 348
“Configure a new APPC Connection to a Tier-3 System” on page 353
“Configure the iPAAServices application onto the application server” on page 394
“Configure a Server for Connections to tier-3 Systems” on page 392
“Configure a new Connection to a Database for use by Applications” on page 237

## Configure a HOD Connection for an Application

Use this procedure to specify the connection details on a HOD Connection to be used by one or more applications for 3270 communication with a tier-3 CICS region or IMS server.

A Component Broker application program that needs to use 3270 communication normally provides a *HOD Connection* that you can use to configure details of the connection. When you load the application into a system management Configuration, you specify the connection details by editing the HOD Connection.

**Prerequisites:**

You only need to complete this task if the connection details have not already been set within the application family package.

This task involves creating and editing a HOD Connection, and relating the Connection to the Application that is to use the Connection.

The application should already have been “Add an Application into a Configuration of your Application Environment” on page 228.

If the HOD Connection already exists, you do not have to create a new one. Likewise, if the relationships already exist, you do not have to create new ones. For example, a HOD Connection and its relationships may have been created automatically from the application’s DDL file when the application was added into the Configuration.

If the HOD Connection was created automatically when the application was added into the Configuration, it may have a default name. You should consider renaming the HOD Connection to something unique and appropriate to the use of the connection.

**To configure a new HOD Connection, complete the following steps:**

1. Expand the Configuration that contains models for your Application and the Servers that it is to run on.
2. To display the HOD Connections provided by the application, complete the following steps:
   a. Expand the Applications folder
   b. Expand the Application that is to use the HOD Connection
   c. Expand the *Provided HOD Connections* folder

      Within this folder you should see a HOD Connection. If not, the application has not provided one.
3. If the application has provided a HOD Connection, and you want to rename it to something unique and appropriate to the use of the connection, complete the following steps:
   a. From the pop-up menu of the HOD Connection, click **Rename**
   b. In the dialog box displayed, type an appropriate, unique, name for the HOD Connection. This name is used for system management purposes only. It is an ASCII string and can contain embedded blanks.
   c. To apply the new name, click the **OK** button in the dialog box.
4. If the application has not provided an HOD Connection, create a new one by completing the following steps. Otherwise, you can edit the HOD Connection as described in step 5.
   a. From the pop-up menu of your Configuration, click **New - HOD Connection**
   b. In the dialog box displayed, type an appropriate, unique, name for the new HOD Connection. This name is used for system management purposes only. It is an ASCII string and can contain embedded blanks.
   c. To create the HOD Connection, in the dialog box click the **OK** button.
   d. To display the HOD Connection, expand the HOD Connections folder.

      Relate the HOD Connection to the Application that is to use it, by completing the following steps:
   e. On the pop-up menu of the HOD Connection, click **Drag**.
   f. On the pop-up menu of the Application that is to use the HOD Connection, click **Configure HOD Connection**. This adds the HOD Connection to the *Provided HOD Connections* folder within the Application (displayed in an earlier step).

   If an application uses several HOD Connections, repeat this step for each other HOD Connection required.

   If several applications use the same HOD connection characteristics, you can configure the same HOD Connection onto each Application. To do this, repeat steps v to vi for each other Application.
5. Edit the HOD Connection to set the characteristics of the connection. You set the characteristics by changing the attributes of the HOD Connection to the values determined in “Configure a new HOD Connection to a Tier-3 System” on page 344 and others according to information provided with the application.

   To edit the HOD Connection’s attributes, complete the following steps:

a. From the pop-up menu of the HOD Connection, click **Edit**. This displays the Object Editor for the HOD Connection.

b. Click the **Main** notebook tab.

c. Set the following attributes to your chosen values:

**create function name**
The name of the function used to create new instances of this Connection

**description**
An optional 256-character text field for you to store any text

**host name**
The TCP/IP host name of the tier-3 system on which the telnet listener is running.

**logon logoff class name**
The name of the class that provides **logon** and **logoff** methods for this HOD Connection, as specified in information provided with the application.

**maximum number of connections**
The maximum number of connection instances allowed for this Connection. If the maximum number of connections is reached, a connection request is delayed for the period specified by the **max wait time** attribute.

The value is an integer equal to or greater than 0; default, 12. A value of 0 (zero) implies no maximum limit.

The corresponding **maximum number of HOD connections** attribute for a server or server group can further limit the maximum number of connections that can exist at the same time. If the server limit is reached before this application-specific limit, a connection request is delayed for the period specified by the application server's **max wait time for HOD connections** attribute.

**maximum wait time**
The maximum time, in milliseconds, that a connection request is delayed if the maximum number connections already exists; default, 300 milliseconds.

A value of 0 (zero) means that the maximum wait time for the server on which application is running should be used. If that corresponding **maximum wait time for HOD connections** attribute of the server is also set to 0 (zero), a delayed connection request will wait indefinitely, until the connection can be created.

**port number**
The port number used by the telnet listener for the tier-3 system; default 23. The value can be an integer port number in the range 0 through 65535.

**version**
(Optional) The version of this HOD Connection

d. When you have finished, click the **OK** button to apply your changes and close the Object Editor window.

6. To verify that your application Configuration is valid, and to apply the changes to the runtime configuration of your application servers, activate the Configuration. To do this, from the pop-up menu of the Configuration click **Activate**.

**Related Concepts**

"HOD Connections to Tier-3 Systems" on page 32

**Related Tasks**

"Configure an ECI Connection for an Application" on page 349
"Configure an APPC Connection to a Tier-3 System for use by Applications" on page 354
"Configure the iPAAServices application onto the application server" on page 394
"Configure a Server for Connections to tier-3 Systems" on page 392

## Configure a new ECI Connection to a Tier-3 System

Use this procedure to configure a connection to a tier-3 CICS region to use the External Call Interface (ECI) of a CICS Client. This connection can be used by one or more applications running on Component Broker application servers.

If you want to configure a different type of connection to a tier-3 system, see the related tasks at the end of this topic.

ECI communication between an application server and a tier-3 CICS region is via a CICS Transaction Gateway and a CICS Client, as shown in the following figure:

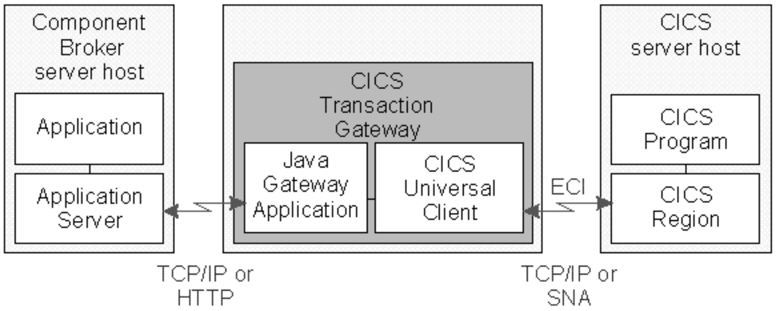


*Figure 69. ECI Connection to a tier-3 CICS Region*

If you want to use an existing CICS Transaction Gateway and CICS Client, you can configure the new ECI connection with their details.

If you want to install and configure a new CICS Transaction Gateway and CICS Client, see the *CICS and IMS Application Adaptor Quick Beginnings*.

For information about configuring communication between CICS Clients and CICS regions, see ″Setting up Client/Server Communication″ in the *CICS Clients: Administration* manual.

You should record your values for the following parameters to be used later when configuring the ECI Connection:

*Table 18.* **Parameters used to Configure an ECI Connection to a Tier-3 System**

| Parameter | Example Value | Notes |
|---|---|---|
| **CICS server name** | CICSSRVA | The name of the CICS region that the ECI connection is to, defined in a server section of the CICS Client initialization file |
| **Java Gateway protocol** | tcp | From the Java Gateway configuration |
| **Java Gateway address** | my.cics.gateway | From the Java Gateway configuration |
| **Java Gateway port** | 8080 | From the Java Gateway configuration |

To configure an ECI connection with a tier-3 CICS region, use the System Manager user interface to complete the following tasks:

1. “Configure an ECI Connection for an Application”.
   Complete this task to specify connection details by editing an ECI Connection model, which is normally created automatically by the application that needs to use the connection.
2. “Configure the iPAAServices application onto the application server” on page 394
   Complete this task only if you need to add the Procedural application adaptor (PAA) services (iPAAServices application) to an application server. These services enable an application server to communicate with tier-3 systems. If the application server already has the iPAAServices application (perhaps added for another tier-3 connection), you do not need to complete this task again.
3. “Configure a Server for Connections to tier-3 Systems” on page 392
   Complete this task only if you need to define the userid and password that an application server uses to connect securely with *any* tier-3 system or if you need to change server-wide connection limits. If you have already specified the userid, password, or server-wide limits, you do not need to complete this task again.

**Related Concepts**

“CICS Transaction Gateway” on page 34
“ECI Connections to Tier-3 Systems” on page 31

**Related Tasks**

Configure a new HOD Connection to a Tier-3 System
Configure a new APPC Connection to a Tier-3 System
“Configure the iPAAServices application onto the application server” on page 394
“Configure a Server for Connections to tier-3 Systems” on page 392
“Configure a new Connection to a Database for use by Applications” on page 237

## Configure an ECI Connection for an Application

Use this procedure to specify the connection details on an ECI Connection to be used by one or more applications for ECI communication with a tier-3 CICS region.

A Component Broker application program that needs to use ECI normally provides an *ECI Connection* that you can use to configure details of the connection. When you load the application into a system management Configuration, you specify the connection details by editing the ECI Connection.

**Prerequisites:**

You only need to complete this task if the connection details have not already been set within the application family package.

This task involves creating and editing an ECI Connection, and relating the Connection to the Application that is to use the Connection.

The application should already have been "Add an Application into a Configuration of your Application Environment" on page 228.

If the ECI Connection already exists, you do not have to create a new one. Likewise, if the relationships already exist, you do not have to create new ones. For example, an ECI Connection and its relationships may have been created automatically from the application's DDL file when the application was added into the Configuration.

If the ECI Connection was created automatically when the application was added into the Configuration, it may have a default name. You should consider renaming the ECI Connection to something unique and appropriate to the use of the connection.

**To configure an ECI Connection, complete the following steps:**

1. Expand the Configuration that contains models for your Application and the Servers that it is to run on.
2. To display the ECI Connections provided by the application, complete the following steps:
   a. Expand the Applications folder
   b. Expand the Application that is to use the ECI Connection
   c. Expand the *Provided ECI Connections* folder

      Within this folder you should see an ECI Connection. If not, the application has not provided one.
3. If the application has provided an ECI Connection, and you want to rename it to something unique and appropriate to the use of the connection, complete the following steps:
   a. From the pop-up menu of the ECI Connection, click **Rename**
   b. In the dialog box displayed, type an appropriate, unique, name for the ECI Connection. This name is used for system management purposes only. It is an ASCII string and can contain embedded blanks.
   c. To apply the new name, click the **OK** button in the dialog box.

   You can then edit the ECI Connection as described in step 5.
4. If the application has not provided an ECI Connection, create a new one by completing the following steps. Otherwise, you can edit the ECI Connection as described in step 5.
   a. From the pop-up menu of your Configuration, click **New - ECI Connection**
   b. In the dialog box displayed, type an appropriate, unique, name for the new ECI Connection. This name is used for system management purposes only. It is an ASCII string and can contain embedded blanks.
   c. To create the ECI Connection, in the dialog box click the **OK** button.
   d. To display the ECI Connection, expand the ECI Connections folder.

Relate the ECI Connection to the Application that is to use it, by completing the following steps:

e. On the pop-up menu of the ECI Connection, click **Drag**.

f. On the pop-up menu of the Application that is to use the ECI Connection, click **Configure ECI Connection**. This adds the ECI Connection to the *Provided ECI Connections* folder within the Application (displayed in an earlier step).

If an application uses several ECI Connections, repeat this step for each other ECI Connection required.

If several applications use the same ECI connection characteristics, you can configure the same Connection model onto each Application model. To do this, repeat steps v to vi for each other Application.

5. Edit the ECI Connection to set the characteristics of the connection. You set the characteristics by changing the attributes of the ECI Connection to the values determined in “Configure a new ECI Connection to a Tier-3 System” on page 348 and others according to information provided with the application.

   To edit the ECI Connection’s attributes, complete the following steps:

   a. From the pop-up menu of the ECI Connection, click **Edit**. This displays the Object Editor for the ECI Connection.
   b. Click the **Main** notebook tab.
   c. Set the following attributes to your chosen values:

      **CICS server name**
      The name of the CICS server (region) that the ECI Connection is to defined in a server section of the CICS Client initialization file.

      **create function name**
      The name of the function used to create new instances of this Connection

      **description**
      An optional 256-character text field for you to store any text

      **gateway address**
      The network address of the CICS Transaction Gateway.

      If the CICS Transaction Gateway is local (installed on the same computer), leave the default (`local`). Otherwise, if a *network* CICS Transaction Gateway is required, type the gateway address in one of the following forms:

      **`tcp://cics.gateway.ip.addr/`**
      The connection uses the CICS Transaction Gateway at TCP/IP address `cics.gateway.ip.addr`. A simple Gateway-specific protocol is used on the connection.

      **`http://cics.gateway.ip.addr/`**
      The connection uses the CICS Transaction Gateway at TCP/IP address `cics.gateway.ip.addr` and the HTTP protocol. The following points should be considered:

      - This protocol is good in environments where an HTTP proxy-firewall is installed.
      - Since HTTP is a non-persistent protocol, performance is an order of magnitude worse that the `tcp:` protocol.

**`auto://cics.gateway.ip.addr`**
An *automatic* `JavaGateway` is to be created. If at runtime the localhost TCP/IP address equals the specified `cics.gateway.ip.addr` address then the connection uses a *local* `JavaGateway`; otherwise it uses a *network* `JavaGateway` for the CICS Transaction Gateway at the TCP/IP address `cics.gateway.ip.addr`.

**`cics.gateway.ip.addr`**
If no protocol is specified, then the connection uses a default protocol of `tcp:` and the CICS Transaction Gateway at TCP/IP address `cics.gateway.ip.addr`.

**logon logoff class name**
The name of the class that provides **logon** and **logoff** methods for this ECI Connection, as specified in information provided with the application.

**maximum number of connections**
The maximum number of connection instances allowed for this Connection. If the maximum number of connections has already been reached, a connection request is delayed for the period specified by the **max wait time** attribute.

The value is an integer equal to or greater than 0; default, 12. A value of 0 (zero) implies no maximum limit.

The corresponding **maximum number of ECI connections** attribute for a server or server group can further limit the maximum number of connections that can exist at the same time. If the server limit is reached before this application-specific limit, a connection request is delayed for the period specified by the application server's **max wait time for ECI connections** attribute.

**maximum wait time**
The maximum time, in milliseconds, that a connection request is delayed if the maximum number connections already exists; default, 300 milliseconds.

A value of 0 (zero) means that the maximum wait time for the server on which application is running should be used. If the corresponding **maximum wait time for ECI connections** attribute of the server is also set to 0 (zero), a delayed connection request will wait indefinitely, until the connection can be created.

**version**
(Optional) The version of this ECI Connection

d. When you have finished, click the **OK** button to apply your changes and close the Object Editor window.

6. To verify that your application Configuration is valid, and to apply the changes to the runtime configuration of your application servers, activate the Configuration. To do this, from the pop-up menu of the Configuration click **Activate**.

### Related Concepts

"CICS Transaction Gateway" on page 34
"ECI Connections to Tier-3 Systems" on page 31

**Related Tasks**

“Configure a HOD Connection for an Application” on page 345
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354
“Configure the iPAAServices application onto the application server” on page 394
“Configure a Server for Connections to tier-3 Systems” on page 392

## Configure a new APPC Connection to a Tier-3 System

Use this procedure to configure a connection to a tier-3 CICS region to use Advanced Program-to-Program Communications (APPC). This connection can be used by one or more applications running on Component Broker application servers.

If you want to configure another type of connection to a tier-3 system, see the related tasks at the end of this topic.

APPC communication between an application server and a tier-3 CICS region is via an IBM Communications Server running on the same host as the application server, as shown in the following figure:

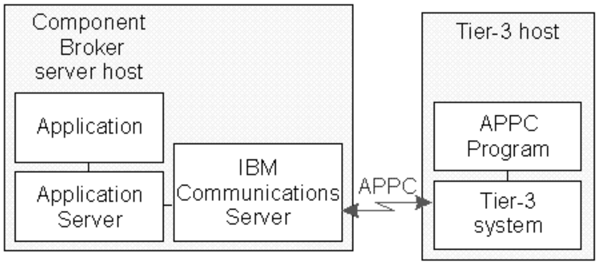


*Figure 70. APPC Connection to a tier-3 CICS Region*

If you want to use an existing IBM Communications Server, you can configure it with details of the new APPC connection.

If you want to install and configure a new IBM Communications Server onto a Component Broker server host, see the *Communications Server Up and Running Guide*.

To configure an ECI connection with a tier-3 CICS region, use the System Manager user interface to complete the following tasks:

1. “Collect information for your SNA configuration” on page 358
   Complete this task to identify SNA details to be used to configure IBM Communications Server and APPC Connections.
2. “Configure Communications Server” on page 367
   Complete this task to configure IBM Communications Server with details identified in the previous task.
3. “Configure VTAM with details of your Component Broker APPC Connections” on page 388
   Complete this task to configure VTAM with details about the APPC connection

so that the so that network control program can pass requests from a Component Broker host (as a SNA node) to VTAM and the remote system beyond.

4. “Configure an APPC Connection to a Tier-3 System for use by Applications”. Complete this task to specify connection details by editing an APPC Connection model, which is normally created automatically by the application that need to use the connection.
5. “Configure the iPAAServices application onto the application server” on page 394
   Complete this task only if you need to add the Procedural application adaptor (PAA) services (iPAAServices application) to an application server. These services enable an application server to communicate with tier-3 systems. If the application server already has the iPAAServices application (perhaps added for another tier-3 connection), you do not need to complete this task again.

**Related Concepts**

“CICS Transaction Gateway” on page 34
“APPC Connections to Tier-3 Systems” on page 33

**Related Tasks**

Configure a new HOD Connection to a Tier-3 System
“Configure a new ECI Connection to a Tier-3 System” on page 348
“Configure the iPAAServices application onto the application server” on page 394
“Configure a new Connection to a Database for use by Applications” on page 237

## Configure an APPC Connection to a Tier-3 System for use by Applications

Use this procedure to configure an APPC connection to a tier-3 System (for example, a CICS region or IMS server) for use by one or more applications running on Component Broker application server.

A Component Broker application program that needs to use APPC communication normally provides an *APPC Connection* that you can use to configure details of the connection. When you load the application into a system management Configuration, you specify the connection details by editing the APPC Connection.

**Prerequisites:**

- You only need to complete this task if an application is to use an APPC connection to a tier-3 system and the characteristics have not already been set within the application family package.
- This task involves creating and editing an APPC Connection, and relating the APPC Connection to Applications that are to use it.

  The application should already have been “Add an Application into a Configuration of your Application Environment” on page 228.

  If the APPC Connection model already exists, you do not have to create a new one. Likewise, if the relationships already exist, you do not have to create new ones. For example, an APPC Connection model and its relationships may have been created automatically from the application’s DDL file when the application was added into the Configuration.

If the APPC Connection model was created automatically when the application was added into the Configuration, it may have a default name. You should consider renaming the APPC Connection model to something unique and appropriate to the use of the connection.

- You should already have collected the information that you need to specify for attributes of the APPC Connection. This is normally done when you "Collect information about local application servers" on page 362.

**To configure an APPC Connection, complete the following steps:**

1. Expand the Configuration that contains models for your Application and the Servers that it is to run on.
2. To display the APPC Connections provided by the application, complete the following steps:
   a. Expand the Applications folder
   b. Expand the Application that is to use the APPC Connection
   c. Expand the *Provided APPC Connections* folder

      Within this folder you should see an APPC Connection. If not, the application has not provided one.
3. If the application has provided an APPC Connection, and you want to rename it to something unique and appropriate to the use of the connection, complete the following steps:
   a. From the pop-up menu of the APPC Connection, click **Rename**
   b. In the dialog box displayed, type an appropriate, unique, name for the APPC Connection. This name is used for system management purposes only. It is an ASCII string and can contain embedded blanks.
   c. To apply the new name, click the **OK** button in the dialog box.

   You can then edit the APPC Connection as described in step 5.
4. If the application has not provided an APPC Connection, create a new one by completing the following steps. Otherwise, you can edit the APPC Connection as described in step 5.
   a. From the pop-up menu of your Configuration, click **New - APPC Connection**.
   b. In the dialog box displayed, type an appropriate, unique, name for the new APPC Connection. This name is used for system management purposes only. It is an ASCII string and can contain embedded blanks.
   c. To create the APPC Connection, click the **OK** button in the dialog box.
   d. To display the APPC Connection, expand the APPC Connections folder.

      Relate the APPC Connection to the Application that is to use it, by completing the following steps:
   e. On the pop-up menu of the APPC Connection, click **Drag**.
   f. On the pop-up menu of the Application that is to use the APPC Connection, click **Configure APPC Connection**. This adds the APPC Connection to the *Provided APPC Connections* folder within the Application (displayed in an earlier step).

   If an application uses several APPC Connections, repeat this step for each other APPC Connection required.

If several applications use the same APPC connection characteristics, you can configure the same Connection model onto each Application model. To do this, repeat steps v to vi for each other Application.

5. Edit the APPC Connection to set the characteristics of the connection. You set the characteristics by changing the attributes of the APPC Connection to the values determined when you “Collect information for your SNA configuration” on page 358 and others according to information provided with the application.

   To edit the APPC Connection’s attributes, complete the following steps:

   a. From the pop-up menu of the APPC Connection, click **Edit**. This displays the Object Editor for the ECI Connection.
   b. Click the **Main** notebook tab.
   c. Set the following attributes to your chosen values:

      **fully-qualified local LU name**
      The SNA name that the Component Broker application server uses to uniquely identify itself when communicating with a tier-3 system. This value, also referred to as the *network-qualified LU name*, must match the value that you identified when you “Collect information about local application servers” on page 362.

      **fully-qualified partner LU name attribute**
      The SNA name that the remote system uses to uniquely identify itself when communicating with Component Broker application servers. This value, also referred to as the *network-qualified LU name*, must match the value that you identified when you “Collect Information About Tier-3 Systems” on page 364.

      **mode name**
      The name used to control how many sessions are created (bound) between the two communicating systems (LUs), and what type of sessions they are. This value must match the value that you identified when you “Collect information about modenames” on page 365 used on APPC connections with Component Broker application servers.

      **remote procedure type**
      The type of tier-3 system that this APPC connection is to; default, CICS DTP:

      **CICS DPL**
      A CICS region using the CICS distributed program logic (DPL) function

      **CICS DTP**
      A CICS region using the CICS distributed transaction processing (DTP) function

      **IMS** An IMS server

      **transaction program name (TPN)**
      If the **remote procedure type** attribute is **CICS DPL**, set this to the name of the transaction to be invoked in the tier-3 CICS region.

      If the value of the **remote procedure type** attribute is **CICS DTP**, leave this attribute to default to ″CPMI″.

      **CICS program name**
      The name of the CICS program to be called in the remote CICS

region for an APPC connection using CICS DPL. Only specify this attribute if the **remote procedure type** attribute is **CICS DPL**; otherwise, this attribute is ignored.

**logon logoff class name**
The name of the class that provides **logon** and **logoff** methods for this APPC Connection, as specified in information provided with the application.

**maximum number of connections**
The maximum number of connection instances allowed for this Connection. If the maximum number of connections has already been reached, a connection request is delayed for the period specified by the **max wait time** attribute.

The value is an integer equal to or greater than 0; default, 12. A value of 0 (zero) implies no maximum limit.

**maximum wait time**
The maximum time, in milliseconds, that a connection request is delayed if the maximum number of connections already exists; default, 300 milliseconds.

A value of 0 (zero) means that a delayed connection request will wait indefinitely, until the connection can be created.

**transaction type**
Whether transactions using this APPC connection make optimistic or pessimistic updates to databases; default, pessimistic:

**pessimistic**
The transaction locks database records when they are read from the database. This prevents other applications from updating the database record while the transaction is running, but reduces overall system concurrency.

**optimistic**
The transaction releases database locks after reading a record. Before updating a record, the transaction checks that the record has not been updated by other transactions. If it has not been updated, the record is locked, the updates are made, and the transaction is committed. If the record has been updated by another transaction, the current transaction is rolled back.

**non_transactional**
This APPC Connection is not used by transactions.

**You are recommended to use the optimistic option**, unless your data changes frequently, there is much update contention for the data among concurrent transactions, or your application must always have the most current value at all times.

**description**
(Optional) A 256-character text field for you to store any text.

**version**
(Optional) The version of this ECI Connection

d. When you have finished, click the **OK** button to apply your changes and close the Object Editor window.

6. To verify that your application Configuration is valid, and to apply the changes to the runtime configuration of your application servers, activate the Configuration. To do this, from the pop-up menu of the Configuration click **Activate**.

### Related Concepts

"Introduction to SNA" on page 471
"APPC Connections to Tier-3 Systems" on page 33
"IBM Communications Server" on page 34

### Related Tasks

"Configure a HOD Connection for an Application" on page 345
"Configure an ECI Connection for an Application" on page 349

## Collect information for your SNA configuration

Use this overview task to collect the information that you need to configure Communications Server for use by Component Broker application servers. You only need to complete this procedure if one or more of your application servers communicate with other systems across an SNA network. You should complete this procedure before starting to configure Communications Server, because you will find SNA configuration much easier with this information ready.

You should record values for the following parameters to be used later when configuring the APPC Connection in Component Broker:

| Parameter | Example Value | Notes |
|---|---|---|
| **fully-qualified** *local LU name* | MYSNANET.CBSERV1 | The *network-qualified LU name* that uniquely identifies the application server in the SNA network. |
| **fully-qualified** *partner LU name* | MYSNANET.CICSESA | The *network-qualified LU name* that uniquely identifies the tier-3 system in the SNA network. |
| **modename** | CB2CICS0 | The name of a modegroup used to define the number of sessions allowed between the application server and the tier-3 system. |

To collect the information that you need to configure Communications Server, complete the following tasks:

1. "Collect information about the local host" on page 359
2. "Collect Information About Remote Hosts" on page 360
3. "Collect information about local application servers" on page 362
4. "Collect Information About Tier-3 Systems" on page 364
5. "Collect information about modenames" on page 365
6. "Match Communications Server names with CICS definitions" on page 366

When you have collected the information you need about the systems that you want to connect across a SNA network, you are ready to configure Communications Server. This is described in "Configure Communications Server" on page 367.

**Related Concepts**

"IBM Communications Server" on page 34
"Introduction to SNA" on page 471
"Connections to Tier-3 Systems" on page 29

**Related Tasks**

"Configure Communications Server" on page 367
"Configure an APPC Connection to a Tier-3 System for use by Applications" on page 354

## Collect information about the local host

Use this procedure to collect the information you need to register a Component Broker host computer within an SNA network. The information is used to configure the Communications Server on that host. You need to complete this procedure only if the host has one or more application servers that communicate with other systems across an SNA network. It forms a stage of "Collect information for your SNA configuration" on page 358 to collect all the information that you need to configure Communications Server for use by Component Broker application servers.

The aim of this procedure is to create a set of information, as shown in the example table Information describing the local host (page 359). You might find it useful to create a similar table with an entry for each of your Component Broker host computers that has one or more application servers that communicate with other systems across an SNA network

*Table 19.* **Example: Information describing the local host**

| *CP Name* | *XID* | *Local Network Name* |
|---|---|---|
| *NT000127* | *05D98765* | *MYSNANET* |

**To collect the information that you need to register a Component Broker host computer within an SNA network, complete the following steps:**

1. Choose a *Control Point (CP) name* for the Component Broker host computer. The CP name is from 1 through 8 ASCII characters selected from the following characters:
   - The first character must be an uppercase alphabetic character (A-Z) or special character (@,#,$).
   - The remaining characters can be uppercase alphanumeric characters (A-Z,0-9) or special characters (@,#,$).

   The CP name is the network name by which the host is known in the SNA network. Therefore, you must ensure that the CP name you use is unique within the SNA network. It must be different from the following names:
   - All other CP names in the network.
   - All other *Logical Unit (LU)* names in the network.

   The administrators of some large SNA networks use a naming convention that prevents name clashes. Alternatively, if it is suitable, you could use the host name of your host computer.

   **Important:** The CP name for the local host must be different from the "Collect information about local application servers" on page 362.

2. Record the *Exchange Identifier (XID)* associated with the CP name. This is an eight digit number which, for Windows NT, usually begins 05D and identifies the *Physical Unit (PU)* associated with the control point. If your application servers communicate with remote systems through VTAM, then you must define the XID to VTAM. Some example VTAM definitions are given in the task to “Configure VTAM with details of your Component Broker APPC Connections” on page 388.
3. You also need to know the name of the SNA network to which the Component Broker host computer (and all the application servers running on it) belongs. The SNA *Network Name* is from 1 through 8 ASCII characters selected from the uppercase letters A through Z and the numbers 0 through 9.

**Related Concepts**

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

**Related Tasks**

“Collect information for your SNA configuration” on page 358
“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Collect Information About Remote Hosts

Use this procedure to collect the information that you need to connect a Component Broker host computer to other hosts in the SNA network (referred to as SNA *nodes*). The information is used to configure the Communications Server on that host. You need to complete this procedure only if the host has one or more application servers that communicate with other systems across an SNA network. It forms a stage of “Collect information for your SNA configuration” on page 358 to collect all the information that you need to configure Communications Server for use by Component Broker application servers.

The aim of this procedure is to create a set of information, as shown in the example Information Describing Remote SNA Nodes (page 360). (The information in the table is for the example SNA network shown in the figure An Example Heterogeneous SNA Network (page 361).) You might find it useful to create a similar table with an entry for each of the remote hosts on which SNA systems need to communicate with application servers running on Component Broker host computers.

**Example: Information Describing Remote SNA Nodes**

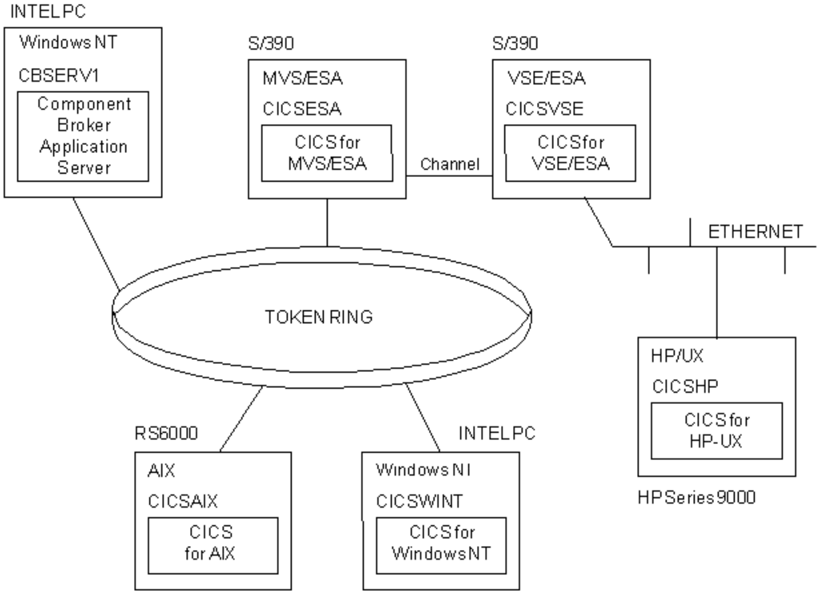
| *Connection Name* | *Link type* | *Hardware Address* | *CP Name* | *Network Name* | *XID* |
|---|---|---|---|---|---|
| *LINKVTAM* | *IBM Token Ring* | *400012345678* | *MYVTAM* | *MYSNANET* | *04300007* |
| *LINKAIX* | *IBM Token Ring* | *10005A4B3C2B* | *AIX00005* | *MYSNANET* | *07101234* |

**To collect the information you need for remote host computers within an SNA network, complete the following steps:**

1. Create a plan of the location of the remote systems that your local application servers communicate with. You need to know which host computers these

systems are running on, and how those computers are connected together. It is often helpful to draw a schematic diagram of the network, such as that shown in the figure An Example Heterogeneous SNA Network (page 361).

**An Example Heterogeneous SNA Network**



Here the Component Broker application server with the LU name **CBSERV1** is to communicate with CICS regions, with the LU names **CICSWINT** (on Windows NT), and **CICSAIX** (on AIX), running on remote hosts connected to the same local area network (LAN). It also is to communicate with two mainframe CICS regions, with the LU names **CICSESA** and **CICSVSE**, and a CICS on Open Systems region called **CICSHP**. The **CICSESA** region is on a host connected to the same LAN as **CBSERV1**. However, **CICSVSE** and **CICSHP** can only be contacted through other hosts. The distinction is important. When you define your connections to Communications Server you only need to include those hosts that can be contacted *directly*. In the example heterogeneous SNA network (page 361), this would be the hosts running **CICSWINT**, **CICSAIX**, and **CICSESA**. Connections to the other hosts (running **CICSVSE** and **CICSHP**) are setup in definitions in the intermediate hosts, so you must consult the owners of such hosts to arrange the appropriate configuration.

After you have determined the hosts that you want Component Broker application servers to connect to, assign a name for the connection to each of these hosts. You will find it useful if you choose names of up to eight ASCII characters that will help you identify the host at the remote end of the connection. This is because the name of the connection appears on the **SNA Node Operations** window where you can start and stop the connection. If the name is meaningful, it makes it much easier to manage your SNA network.

2. Record the type of network that is used to connect the hosts. Typical network types are:
   - IBM Token Ring LAN
   - Ethernet LAN
   - SDLC line
   - X.25

3. Collect information about the remote host’s node. This is similar to the information you collected about your local host’s node in “Collect information about the local host” on page 359 and includes the following parameters for the remote host:
   - Control Point (CP) name
   - Exchange Identifier (XID)
4. Record the hardware (MAC) address of each remote host. To determine the hardware address, use a method appropriate to the type of remote host; for example:
   - **Windows NT**. You can view the hardware address under the **Transports** section of the **Network** applet of its **NT Diagnostics tool** application.
   - **AIX** You can display the address by running the **netstat -v** command on the AIX host.
   - **A mainframe host**. The address you need is probably that of a network controller handling the network traffic on behalf of the mainframe. Ask the SNA Network Administrator for the information you need.
   - **A remote host connected using TCP/IP** To determine the hardware address, complete the following steps:
     a. Use the *ping* command at your local NT host to contact the remote host.
     b. Use the **arp -a** command to list the addresses that the arp (address resolution protocol) knows about.

     For example, the following commands were used to determine the hardware address (10-00-5a-4b-3c-2b)of the host aix5:

```
C:\>ping aix5
Pinging aix5.cicsland.com [1.23.45.67] with 32 bytes of data:
Reply from 1.23.45.67: bytes=32 time<10ms>arp -a
Interface: 1.23.45.987
  Internet Address      Physical Address      Type
  1.23.4.6              08-00-70-81-92-03     dynamic
  1.23.45.67            10-00-5a-4b-3c-2b     dynamic
```

**Related Concepts**

“Collect information for your SNA configuration” on page 358
“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

**Related Tasks**

“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Collect information about local application servers

Use this procedure to collect the information you need to configure an APPC Connection for use by Component Broker application server within an SNA network. The information is used to configure the Communications Server on that host. You need to complete this procedure only if you have one or more application servers that use APPC to communicate with other systems across an SNA network. It forms a stage of “Collect information for your SNA configuration” on page 358 to collect all the information that you need to configure Communications Server for use by Component Broker application servers.

The aim of this procedure is to create a set of information, as shown in the example table Information describing application servers (page 363). You might find it useful to create a similar table with an entry for each of your Component Broker application servers that communicate with other systems across an SNA network.

**Example: Information describing application servers**

| *Application Server Name* | *Local LU Name* |
|---|---|
| *My Server 1* | *CBSERV1* |
| *My Server 2* | *CBSERV2* |

**To collect the SNA information for your application servers, complete the following step:**

1. Choose an LU name for each application server. The Communications Server uses the LU name to uniquely identify the application server. An LU name is from 1 through 8 ASCII characters selected from the following characters:
   - The first character must be an uppercase alphabetic character (A-Z) or special character (@,#,$).
   - The remaining characters can be uppercase alphanumeric characters (A-Z,0-9) or special characters (@,#,$).

   **Important:**
   - Each LU name that you choose for an application server must be different from the following names:
     – The CP name for the local host (See “Collect information about the local host” on page 359.)
     – All other CP names in the network.
     – All other Logical Unit (LU) names in the network; for example, LU names used by CICS clients running on the same local host.
   - The CP name for an application server *must not* be defined to the Communications Server. Component Broker automatically creates the local LU names for its application servers in the Communications Server, when you first activate a Configuration that defines those servers.

The example in the table Information describing application servers (page 363) shows LU names for two application servers, **My Server 1** and **My Server 2** (as identified through the System Manager user interface). The application server **My Server 1** uses the LU name **CBSERV1** and the application server **My Server 2** uses the LU name **CBSERV2**.

### Related Concepts

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

### Related Tasks

“Collect information for your SNA configuration” on page 358.
“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Collect Information About Tier-3 Systems

Use this procedure to collect the information that you need to know about the remote systems that your Component Broker applications servers communicate with across an SNA network. These remote systems are commonly referred to as *Partner LUs*. The information is used to configure the Communications Server on the same host as an application server. You need to complete this procedure only if you have one or more application servers that communicate with other systems across an SNA network. It forms a stage of “Collect information for your SNA configuration” on page 358 to collect all the information that you need to configure Communications Server for use by Component Broker application servers.

The aim of this procedure is to create a set of information, as shown in the example tables Information describing tier-3 systems (page 364). You might find it useful to create similar tables with an entry for each tier-3 system that your Component Broker application servers communicate with across an SNA network.

**Example: Information describing tier-3 systems**

| *LU Name* | *Network Name* | *CP Name of Remote Host* |
|---|---|---|
| *CICSESA* | *MYSNANET* | *MYSNANET.MYVTAM* |
| *CICSVSE* | *MYSNANET* | *MYSNANET.MYVTAM* |
| *CICSAIX* | *MYSNANET* | |
| *CICSHP* | *MYSNANET* | *MYSNANET.MYVTAM* |

**To collect the SNA information for tier-3 systems, complete the following steps:**

1. Record the LU name of each remote system. Mainframe-based CICS regions use their APPLID as their LU name, and may also refer to it as a *NETNAME*. The LU name for CICS on Open Systems regions is specified in their local SNA product’s Local LU definition. The LU name for a CICS for OS/400 region is found in the APPL parameter of the ADDCICSSIT command.
2. Record the name of the SNA network to which each remote system belongs. An SNA network name is up to eight characters long selected from the following characters:
   - The first character must be an uppercase alphabetic character (A-Z) or special character (@,#,$).
   - The remaining characters can be uppercase alphanumeric characters (A-Z,0-9) or special characters (@,#,$).
3. If the remote host used to connect to the tier-3 system does not support “Introduction to SNA” on page 471, then record the “Collect Information About Remote Hosts” on page 360.

The table Information describing remote CICS regions (page 364) shows example data collected for remote CICS regions.

### Related Concepts

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

### Related Tasks

“Collect information for your SNA configuration” on page 358.

"Configure Communications Server" on page 367
"Configure an APPC Connection to a Tier-3 System for use by Applications" on page 354

## Collect information about modenames

Use this procedure to collect the information that you need to know about the *modenames* used on the SNA connections between your Component Broker application servers and remote SNA systems. The information is used to configure the APPC Connections used by applications running on the application servers. You need to complete this procedure only if you have one or more application servers that use APPC to communicate with other systems across an SNA network. It forms a stage of "Collect information for your SNA configuration" on page 358 to collect all the information that you need to configure SNA communications for use by Component Broker application servers.

The aim of this procedure is to choose one or more modenames for use by APPC connections. By default Component Broker defines default parameters to the modenames, but you can choose your own values for those parameters, as shown in the example table Information about modegroups (page 365). You might find it useful to create similar tables with an entry for each modegroup that your Component Broker application servers use to communicate across an SNA network.

**Information about modegroups**

| *Modename* | *Max Sessions* | *Min Winners* |
|---|---|---|
| *CB2CICS0* | *20 (default)* | *10 (default)* |

A modename is the name of a *modegroup*, or *mode*, and is up to eight characters long, made up of upper case letters (A-Z) and numbers (0-9). Modegroups are used to define the number of sessions allowed between two systems (referred to as an *LU-pair*). This is important because each session can only support one APPC request at a time, so the number of sessions between an LU-pair affects the number of concurrent intersystem requests they can process. Being able to control the number of sessions between an LU-pair enables you to control the network traffic and prevent one system from flooding another with requests.

Different LU-pairs can share modegroups. The session limits defined in a modegroup apply independently to each pair of LUs that uses it.

An LU-pair can use more than one modegroup (on different APPC Connections); you would do this if you wanted to give one group of transactions priority over another. For example, if you had an LU-pair that handled both fast intersystem requests for interactive users and slower intersystem requests that send or receive large amounts of data, then you may choose to create two modegroups; one for the interactive requests (for example, CB2CICS0) and the other for the slower data transfer requests (for example, CB2CICS1). You could then set the number of sessions in CB2CICS1 lower than the number of sessions in CB2CICS0, thus allowing more interactive requests to run concurrently than the slower data transfer requests.

The modegroup also defines the characteristics of the sessions that belong to the connection. The most important characteristic is *contention*. Each session in the modegroup must always give priority to one of the communicating systems if both systems simultaneously attempt to start an intersystem request on the same session. The system that is always given priority on the session is called the *contention winner* and other system is called the *contention loser*.

Component Broker automatically creates modegroups needed with the default of 10 sessions contention winners. If you need to use a different value, you must create your own modegroup; if so, consider the following points:

- Ensure that each system has at least one contention winner.
- The modegroups used on a connection must be defined with consistent session limits on both systems.
- Do not attempt to define or use the system-defined modegroups that are reserved for use by SNA and APPN management programs. (For more information, see SNASVCMG and CPSVCMG modegroups (page 366).)
- If Communications Server uses VTAM to communicate with the remote system, then VTAM requires a MODEENT definition for the modename in its tables. For an example of a MODEENT definition, see “Configure VTAM with details of your Component Broker APPC Connections” on page 388.

### SNASVCMG and CPSVCMG modegroups

If you have had experience of SNA networks you may have seen that all LU-pairs using parallel (more than one) sessions have a modegroup called *SNASVCMG*. This modegroup is not included in the list above. This is because SNASVCMG is a system-defined modegroup which is reserved for use by SNA management programs. You should not attempt to define this modegroup in Communications Server, or to use it for your intersystem requests.

APPN systems also use a modegroup called *CPSVCMG*. This is reserved for APPN management programs. Again, you should not attempt to define this modegroup, or to use it for your intersystem requests.

**Related Concepts**

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

**Related Tasks**

“Collect information for your SNA configuration” on page 358.
“Configure Communications Server” on page 367
“Configure VTAM with details of your Component Broker APPC Connections” on page 388
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

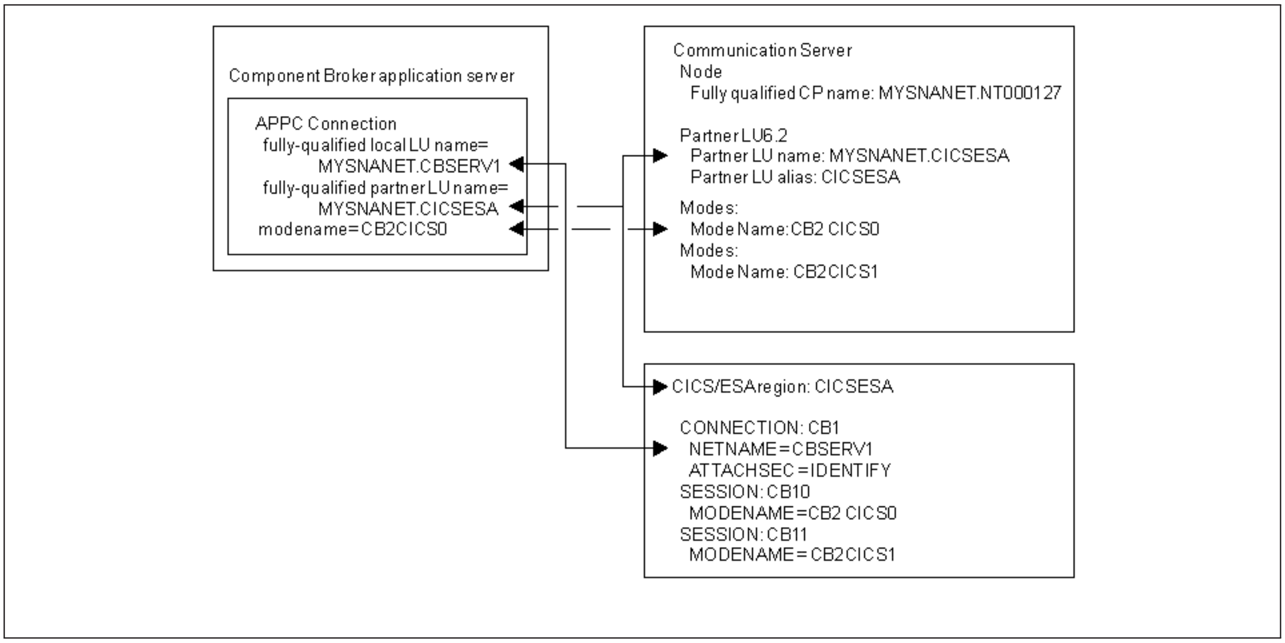
## Match Communications Server names with CICS definitions

Several of the parameters you use to configure Communications Server must match the parameters you use to configure the local Component Broker APPC Connection and the remote partner system.

To ensure that the parameters match, you might find it convenient to draw a schematic diagram of the communicating systems, and annotate the diagram with the parameter values.

For example, the following figure illustrates the matching parameters when a Component Broker application server uses APPC to connect to CICS for MVS/ESA across an SNA network:

*Table 20.* **Matching Communications Server parameters with CICS for MVS/ESA**



**Related Concepts**

"IBM Communications Server" on page 34
"Introduction to SNA" on page 471
"Connections to Tier-3 Systems" on page 29

**Related Tasks**

"Collect information about the local host" on page 359
"Collect information for your SNA configuration" on page 358.
"Configure Communications Server"
"Configure an APPC Connection to a Tier-3 System for use by Applications" on page 354

## Configure Communications Server

Use this overview procedure to configure Communications Server to enable your Component Broker application servers to communicate with other systems across an SNA network.

**Prerequisites:**

- You should have already installed the IBM Communications Server on each host where there is a Component Broker application server that is to connect to a SNA network.

  WIN If your host will be communicating over a Local Area Network (LAN) you must select the IEEE 802.2 IBM LLC2 protocol interface when you install the IBM Communications Server for Windows NT. The *Communications Server for Windows NT Up and Running Guide* explains how to install IBM Communications Server for Windows NT.

- Before starting to configure Communications Server, you should have collected the information that you need, as described in “Collect information for your SNA configuration” on page 358.
- You should be familiar with the concepts and terminology of SNA, as outlined in “Introduction to SNA” on page 471.

WIN **To configure Communications Server for Windows NT**, use the **SNA Node Configuration** application to complete the following tasks:

1. “Configure the Local Host in Communications Server” on page 369
2. “Define links from Communications Server” on page 371
3. “Define Your own Modegroups in Communications Server” on page 375 (optional)
4. “Define Remote Systems to Communications Server” on page 378

AIX **To configure Communications Server for AIX**, use the **xsnaadmin** application to complete the following tasks:

1. “Configure the Local Host in Communications Server” on page 380
2. “Define links from Communications Server” on page 382
3. Define modenames in Communications Server″ (optional)
4. “Define Remote Systems to Communications Server” on page 387

The examples in these topics are not designed to cover every aspect of configuring Communications Server. Only those fields that are particularly relevant to configuring the Communications Server for Component Broker are discussed. Many of the other fields can be left to take their default value. If you require more information, you can see a description of each field by selecting the **help** button on the configuration window where the field appears.

Component Broker automatically configures Communications Server for APPC Connections used by application servers on the same host, based on the information you define in APPC Connections within your system management Configuration. To define that information, see “Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354.

If your application servers connect to mainframe-based systems such as CICS for MVS/ESA, CICS/ESA, CICS/MVS, and CICS/VSE regions, you may need to configure the Virtual Telecommunications Access Method (VTAM). This is to enable VTAM to pass requests from a Component Broker host (as a SNA node) to the mainframe-based systems. For information about updating VTAM, see “Configure VTAM with details of your Component Broker APPC Connections” on page 388.

### Related Concepts

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

### Related Tasks

“Collect information for your SNA configuration” on page 358.
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

page 354
“Configure VTAM with details of your Component Broker APPC Connections” on page 388

## Configure the Local Host in Communications Server

WIN Use this procedure to define, in the IBM Communications Server for Windows NT, your local host computer (which supports a Component Broker application server) to SNA.

It forms part of the overview task to “Configure Communications Server” on page 367 for connection between Component Broker application servers and remote systems over an SNA network.

**Prerequisites:**

- You should have already collected the information about the local host that you need to specify in this procedure. If you have not collected that information, see “Collect information about the local host” on page 359.
- This procedure uses the following values as an example of a local host:

| *CP Name* | *XID* | *Local Network Name* |
|---|---|---|
| *NT000127* | *05D98765* | *MYSNANET* |

For a description of these values, see “Collect information about the local host” on page 359.

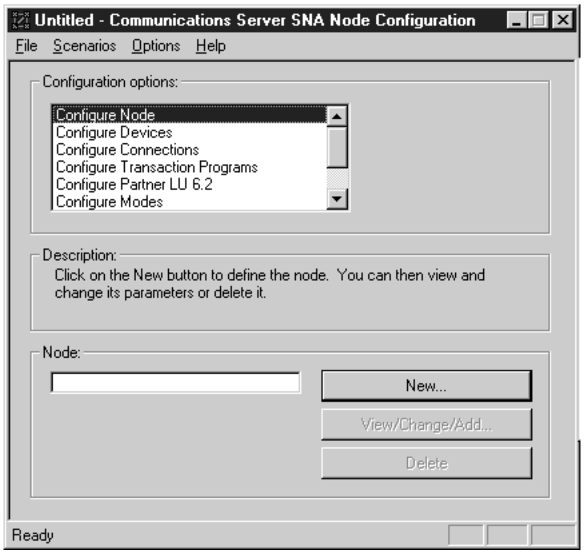
- This procedure uses the SNA Configuration tool provided with Communications Server.

**To configure your local host in Communications Server, complete the following steps:**

1. Display the **SNA Node Configuration** application. The SNA Configuration tool will then request that you either open an existing configuration file or create a new file.
2. On the first panel, click **Scenarios - CPIC, APPC or 5250 Emulation**.

   This displays the panel shown in the figure SNA Node Configuration panel (page 369).

   **SNA Node Configuration panel**

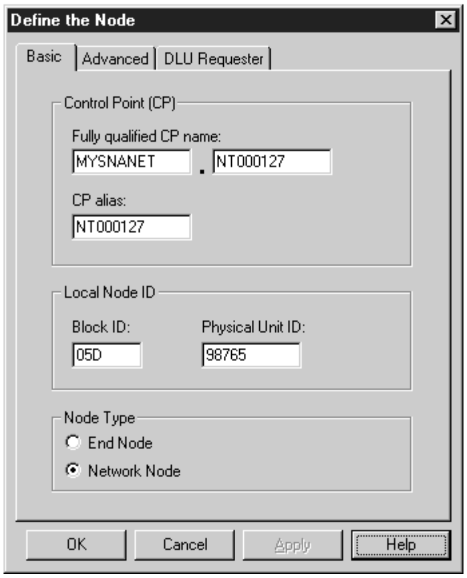


Description:
Click on the New button to define the node. You can then view and change its parameters or delete it.

3. Click **Configure Node**
4. Click the **New** button.

   This displays the panel, shown in Define the Node panel (page 370), that you use to specify the SNA information about your local host.

   **Define the Node panel**



Fully qualified CP name: MYSNANET . NT000127

CP alias: NT000127

Block ID: 05D

Physical Unit ID: 98765

Node Type
○ End Node
● Network Node

5. Under the **Fully qualified CP name**, enter the SNA network name in the first box and the CP name in the second.
6. Enter the CP name in the **CP alias** field.
7. Enter the XID under **Block ID Physical Unit ID**.
8. Select the “Introduction to SNA” on page 471 for the node you are defining.
9. Leave all other options unchanged and click the **OK** button.

**Related Concepts**

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

**Related Tasks**

“Collect information about the local host” on page 359
“Collect information for your SNA configuration” on page 358.
“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Define links from Communications Server

WIN Use this procedure to define, in the IBM Communications Server for Windows NT, links to remote hosts in the SNA network.

It forms part of the overview task to “Configure Communications Server” on page 367 for connection between Component Broker application servers and remote systems over an SNA network.

**Prerequisites:**

- You should have already collected the information about the remote hosts that you need to specify in this procedure. If you have not collected that information, see “Collect Information About Remote Hosts” on page 360.
- This procedure uses the following values as an example:

| *Connection Name* | *Link type* | *Hardware Address* | *CP Name* | *Network Name* | *XID* |
|---|---|---|---|---|---|
| *LINKVTAM* | *IBM Token Ring* | *400012345678* | *MYVTAM* | *MYSNANET* | *04300007* |

  For a description of these values, see “Collect Information About Remote Hosts” on page 360.

- This procedure uses the SNA Configuration tool provided with Communications Server.

**To configure links to remote hosts in Communications Server, complete the following two stages:**

1. Configure a data link control (DLC) device driver (page 371) to control the use of the network connection to the host on which the Communications Server is running.
2. Configure connections between the local host and remote hosts (page 373).

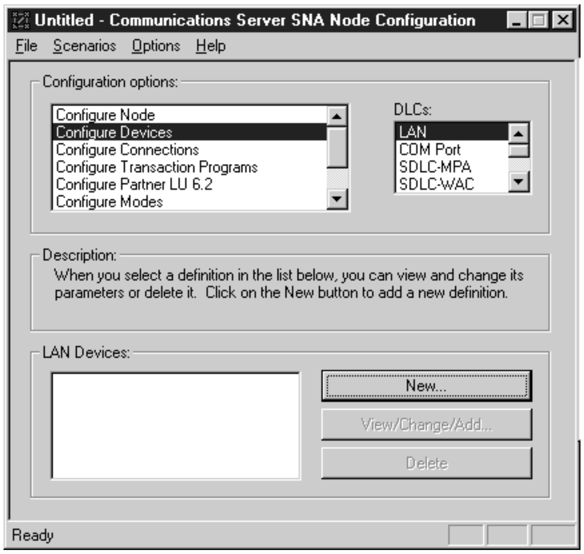
### Configure a data link control (DLC) device driver

To configure a DLC device driver, complete the following steps:

1. Display the **SNA Node Configuration** application.
2. On the first panel, select **Configure Devices**.
3. Select the relevant DLC type you require (for example, select **LAN** if your host is using a LAN).

   This displays the panel shown in the figure SNA Node Configuration, Devices panel (page 371).

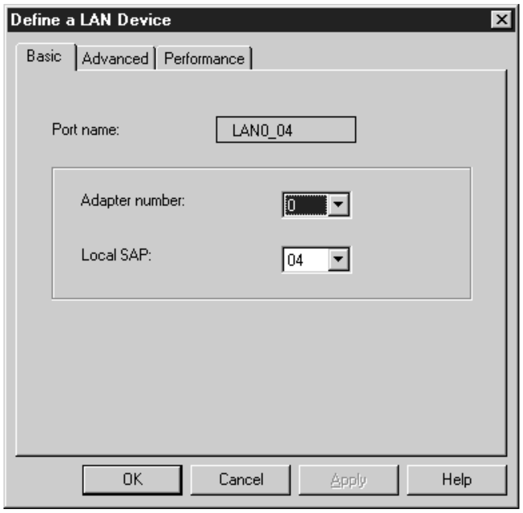
   **SNA Node Configuration, Devices panel**



When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

4. Click the **New** button, to display the Define a LAN Device panel shown in the figure Define a LAN Device panel (page 4).

   **Define a LAN Device panel**

   Port name: LAN0_04

   Adapter number: 0

   Local SAP: 04

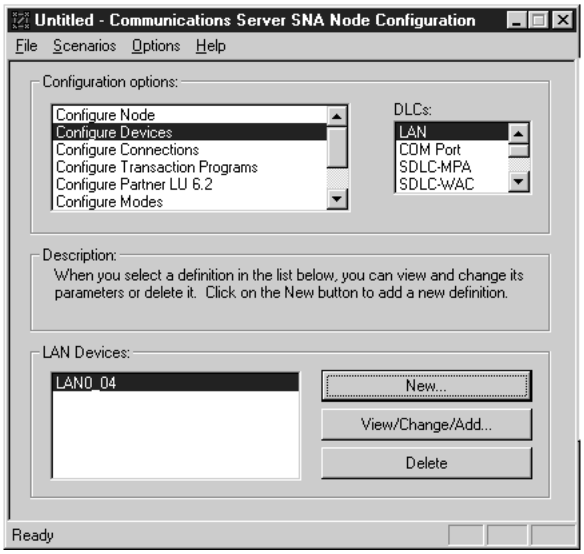
   If the adapter number is blank (not 0), you either have not installed a DLC device driver, or you have not rebooted your machine since the installation. If needed, to install the LLC2 Protocol DLC device driver that can be used for the LAN, complete the following steps:

   a. Switch to the directory where Communications Server was installed;for example **C:\IBMCS**
   b. Run the following program: **INLLC40.HLP**

5. Click the **OK** button to add the DLC device.

   The device will appear in the **SNA Node Configuration** panel, as shown in the figure SNA Node Configuration, with added LAN device (page 372).

   **SNA Node Configuration, with added LAN device**



When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

LAN Devices: LAN0_04

### Configure connections between the local host and remote hosts

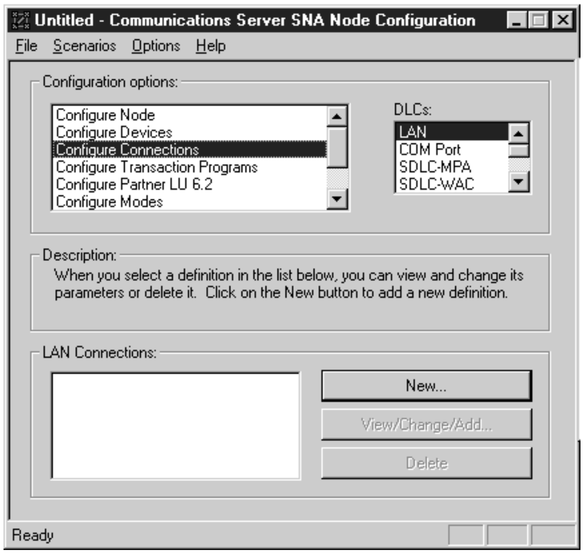
After you have configured a DLC device driver, you can connect your local host to remote hosts by configuring connection definitions.

To configure a connection to a remote host, complete the following steps::

1. Display the **SNA Node Configuration** tool.
2. Select **Configure Connections**.
3. Select the relevant DLC type you require (for example, select **LAN** if your machine is using a LAN).

   This displays the panel shown in the figure SNA Node Configuration, Connection panel (page 373).

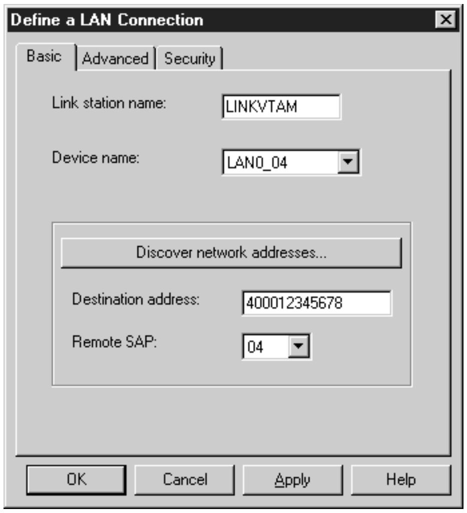
   **SNA Node Configuration, Connection panel**

   When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

4. Click the **New** button, to display the panel shown in the figure Define a LAN Connection panel (page 4).

   **Define a LAN Connection panel**



Link station name: LINKVTAM

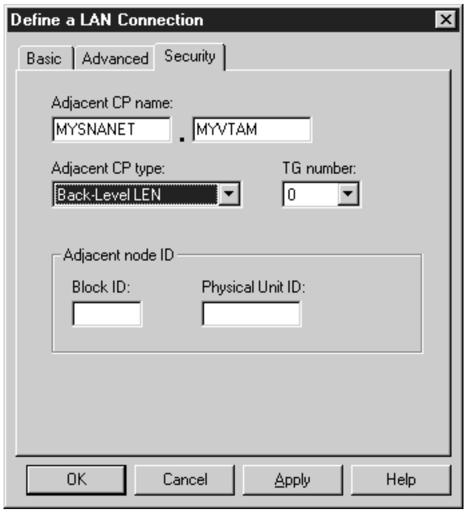
Device name: LAN0_04

Destination address: 400012345678

Remote SAP: 04

5. Type the link station name (for example, LINKVTAM).
6. Type the remote machine address in the destination address.
7. Optionally, you can also configure information about the remote host in the **Security** panel, which is shown in the figure Define a LAN Connection, Security panel (page 7), so you are sure your machine is connecting to the correct machine.

**Define a LAN Connection, Security panel**



Adjacent CP name: MYSNANET . MYVTAM

Adjacent CP type: Back-Level LEN

TG number: 0

Adjacent node ID

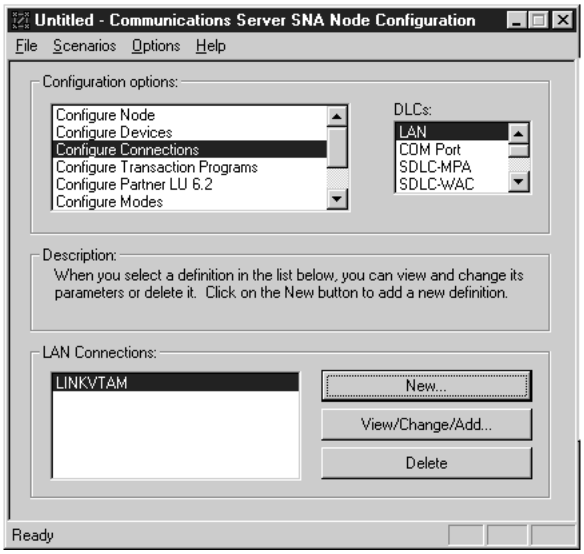
Block ID:

Physical Unit ID:

Click the **OK** button to add the link.

The connection will appear in the **SNA Node Configuration** panel, as shown in the figure SNA Node Configuration, with LINKVTAM connection added (page 374).

**SNA Node Configuration, with LINKVTAM connection added**



When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

**Related Concepts**

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

**Related Tasks**

“Collect Information About Remote Hosts” on page 360
“Collect information for your SNA configuration” on page 358.
“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Define Your own Modegroups in Communications Server

WIN Use this procedure to define, in the IBM Communications Server for Windows NT, your own modegroups for connections across a SNA network. Normally, you do not need to use this procedure, because Component Broker automatically creates the modegroups it needs, based on information that you define when you “Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354.

You should consider completing this task only if you want your connections to use non-default values for the number of sessions or contentions winners.

This procedure forms part of the overview task to “Configure Communications Server” on page 367 for connection between Component Broker application servers and remote systems over an SNA network.

**Prerequisites:**

- You should have already collected the information about the remote hosts that you need to specify in this procedure. If you have not collected that information, see “Collect information about modenames” on page 365.

- This procedure uses the following values as an example:

| *Modename* | *Max Sessions* | *Min Winners* |
|---|---|---|
| *CB2CICS0* | *10* | *5* |

  The values that Component Broker uses by default are:

| *Modename* | *Max Sessions* | *Min Winners* |
|---|---|---|
| *your_modename* | *20* | *10* |

  Where *your_modename* is the modename that you have chosen for the modegroup used by an APPC Connection.

  For a description of these values, see “Collect information about modenames” on page 365.

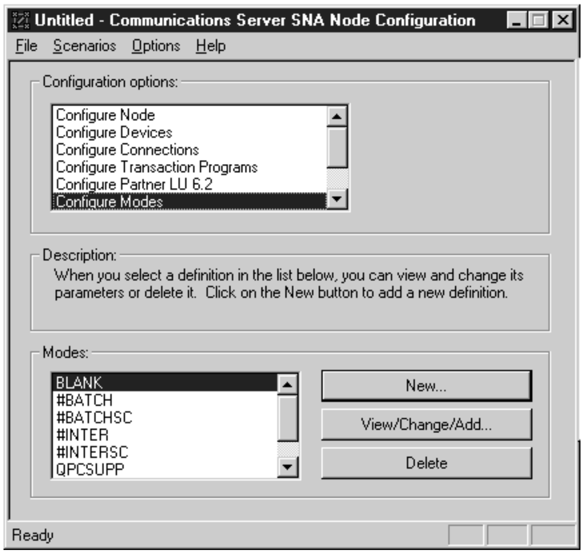
- This procedure uses the SNA Configuration tool provided with Communications Server.

**To configure your own modegroups (also known as *modes*) in Communications Server, complete the following steps:**

1. Display the **SNA Node Configuration** application.
2. On the first panel, select **Configure Modes**.

   This displays the panel shown in the figure SNA Node Configuration, Mode panel (page 376).

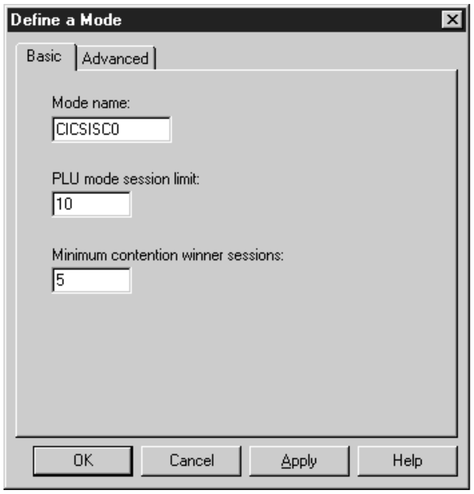
   **SNA Node Configuration, Mode panel**



When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

3. Click the **New** button to display the panel shown in the figure Define a Mode panel (page 3).

   **Define a Mode panel**



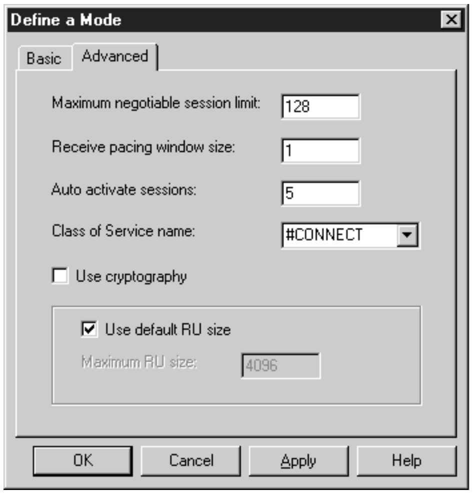
Mode name: CICSISC0

PLU mode session limit: 10

Minimum contention winner sessions: 5

4. Enter the modename (for example, CB2CICS0).
5. Under **PLU mode session limit**, type the maximum number of sessions (for each connection).
6. Under **Minimum contentions winner sessions**, type the minimum number of contention winner sessions (at least 1, and usually about half of the maximum entered above).
7. Optionally, you can change the performance characteristics of the sessions that belong to the mode. To do this, complete the following steps:
    a. Click the **Advanced** tag, to display the panel shown in the figure Define a Mode, Advanced panel (page 7.a).

    **Define a Mode, Advanced panel**

    Maximum negotiable session limit: 128

    Receive pacing window size: 1

    Auto activate sessions: 5

    Class of Service name: #CONNECT

    □ Use cryptography

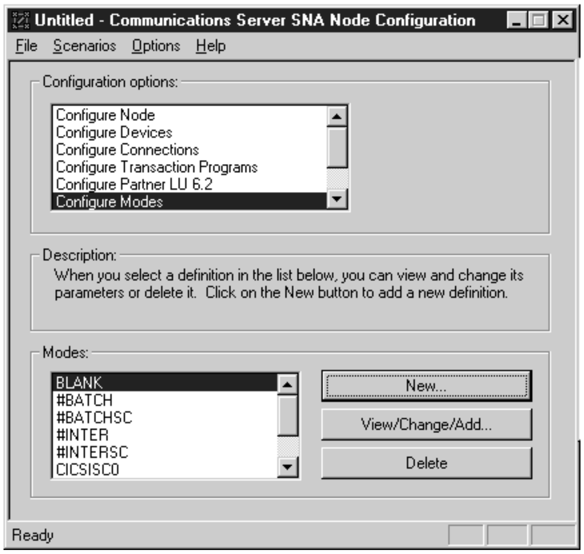
    ☑ Use default RU size

    Maximum RU size: 4096

    b. The default values should be sufficient for an initial configuration. However, better performance can be achieved by matching these values to those defined in the remote LU. If you wish to autoactivate some of your sessions on startup, enter the number to be activated in **Auto activate sessions**.
8. After you have entered the values you require for the modegroup, select the **OK** button to add the modegroup to the Communications Server configuration.

The new modename appears in the list of modenames displayed on the main window. This is shown in the figure SNA Node Configuration, with added Mode (page 377).

**SNA Node Configuration, with added Mode**


Untitled - Communications Server SNA Node Configuration
File Scenarios Options Help
New...
View/Change/Add...
Delete
Ready

Configuration options:

Configure Node
Configure Devices
Configure Connections
Configure Transaction Programs
Configure Partner LU 6.2
Configure Modes

Description:
When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

Modes:

BLANK
#BATCH
#BATCHSC
#INTER
#INTERSC
CICSISC0

### Related Concepts

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

### Related Tasks

“Collect information about modenames” on page 365
“Collect information for your SNA configuration” on page 358.
“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Define Remote Systems to Communications Server

WIN Use this procedure to define, in the IBM Communications Server for Windows NT, the remote systems that your Component Broker application servers communicate with across an SNA network.

This procedure forms part of the overview task to “Configure Communications Server” on page 367 for connection between Component Broker application servers and remote systems over an SNA network.

**Prerequisites:**

- You should have already collected the information about the remote systems that you need to specify in this procedure. If you have not collected that information, see “Collect Information About Tier-3 Systems” on page 364.
- This procedure uses the following values as an example:

| *LU Name* | *Network Name* | *CP Name of Remote Host* |
|---|---|---|
| *CICSESA* | *MYSNANET* | *MYSNANET.MYVTAM* |

For a description of these values, see “Collect Information About Tier-3 Systems” on page 364.

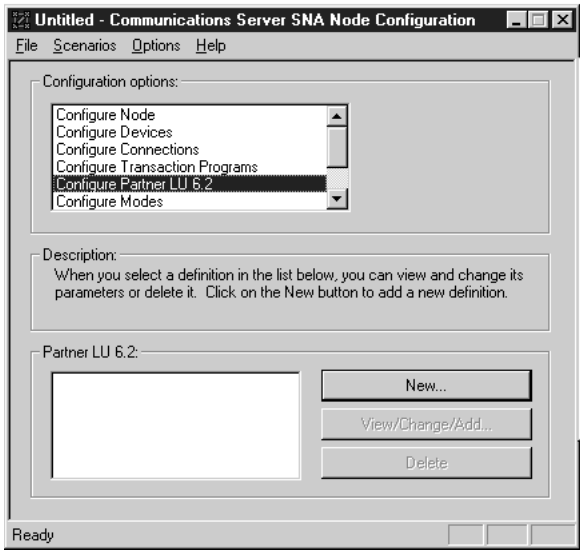
- This procedure uses the SNA Configuration tool provided with Communications Server.
- If you do not define the partner LU name for the remote system, Component Broker will do this automatically. However, in that case, the CP name will be blank, which is alright if the remote system supports “Introduction to SNA” on page 471. If the remote system does not support APPN, then you should define the partner LU name, se described in this topic.

**To define the remote systems that your application servers communicate with, complete the following steps:**

1. Display the **SNA Node Configuration** application.
2. On the first panel, select **Configure Partner LU6.2**.

   This displays the panel shown in the figure SNA Node Configuration, Partner LU 6.2 panel (page 379).

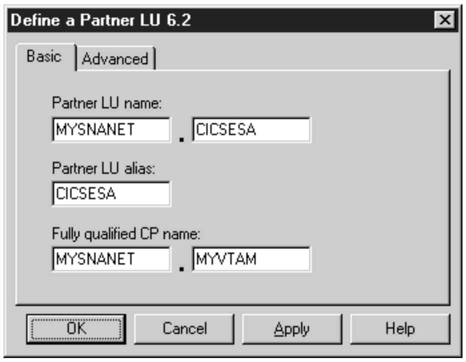
   **SNA Node Configuration, Partner LU 6.2 panel**



When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

3. Click the **New** button, to display the panel shown in the figure Define a Partner LU 6.2 panel (page 3)

   **Define a Partner LU 6.2 panel**



Partner LU name: MYSNANET . CICSESA

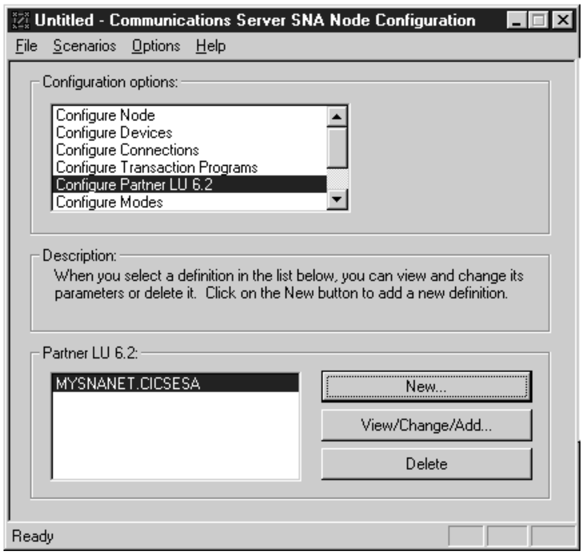
Partner LU alias: CICSESA

Fully qualified CP name: MYSNANET . MYVTAM

4. Enter the fully-qualified partner LU name (MYSNANET.CICSESA).
5. Enter the partner LU alias (CICSESA).
6. Enter the fully qualified CP name of the remote machine.
7. Click the **OK** button, to add the partner LU to the Communications Server configuration.

The new partner LU appears in the list of LU names displayed on the main window. This is shown in the figure SNA Node Configuration panel, with added Partner (page 380).

**SNA Node Configuration panel, with added Partner**



Configuration options:

- Configure Node
- Configure Devices
- Configure Connections
- Configure Transaction Programs
- Configure Partner LU 6.2
- Configure Modes

Description:
When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

Partner LU 6.2:
MYSNANET.CICSESA

**Related Concepts**

"IBM Communications Server" on page 34
"Introduction to SNA" on page 471
"Connections to Tier-3 Systems" on page 29

**Related Tasks**

"Collect information about the local host" on page 359
"Collect information for your SNA configuration" on page 358.
"Configure Communications Server" on page 367
"Configure an APPC Connection to a Tier-3 System for use by Applications" on page 354

## Configure the Local Host in Communications Server

AIX Use this procedure to define, in the IBM Communications Server for AIX, your local host computer (which supports a Component Broker application server) to SNA.

It forms part of the overview task to "Configure Communications Server" on page 367 for connection between Component Broker application servers and remote systems over an SNA network.

**Prerequisites:**

- You should have already collected the information about the local host that you need to specify in this procedure. If you have not collected that information, see "Collect information about the local host" on page 359.

- This procedure uses the following values as an example of a local host:

| *CP Name* | *XID* | *Local Network Name* |
|---|---|---|
| *NT000127* | *05D98765* | *MYSNANET* |

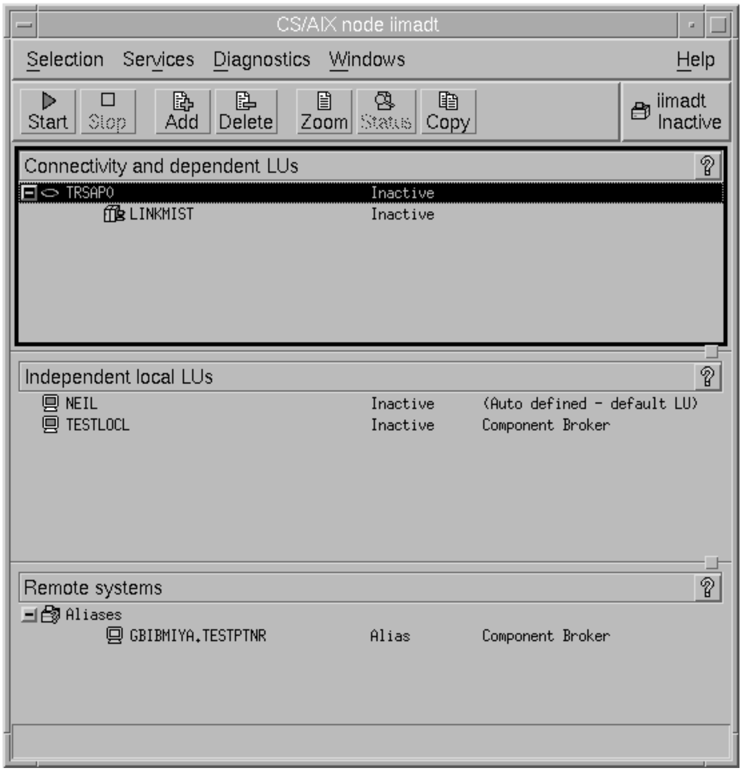
For a description of these values, see “Collect information about the local host” on page 359.

- This procedure uses the SNA Configuration tool provided with Communications Server.

**To configure your local host in Communications Server, complete the following steps:**

1. Display the **xsnaadmin** application. This displays the **SNA Administration** panel, as shown in the figure SNA Administration panel (page 1).

   **SNA Administration panel**

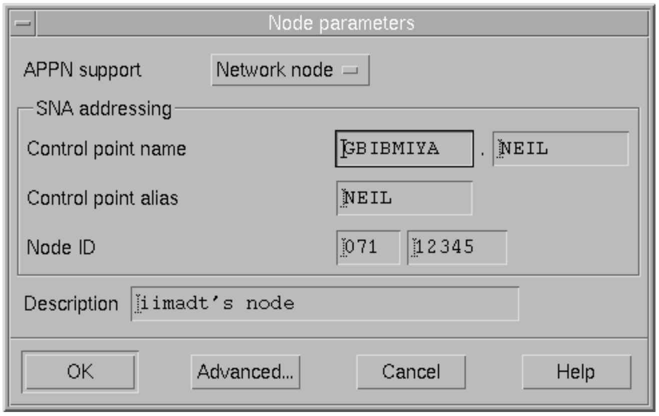
   ```
   TRSAP0                      Inactive
         LINKMIST              Inactive

   NEIL                        Inactive    (Auto defined - default LU)
   TESTLOCL                    Inactive    Component Broker

   Aliases
         GBIBMIYA.TESTPTNR     Alias       Component Broker
   ```

   Select either to open an existing configuration file or create a new file.

2. On the first panel, click **Services - Configure Node Parameters**.

   This displays the panel shown in the figure SNA Node Parameters panel (page 381).

   **SNA Node Parameters panel**



APPN support: Network node

SNA addressing

Control point name: GBIBMIYA . NEIL

Control point alias: NEIL

Node ID: 071 12345

Description: iimadt's node

3. For **APPN support**, select the “Introduction to SNA” on page 471 for the node you are defining.
4. For **Control Point name**, enter the SNA network name in the first box and the Control Point name in the second.
5. Enter the Control Point name in the **Control point alias** field.
6. Enter the XID in the **Node ID** field.
7. Optionally, type a description of the node in the **Description** field.
8. Leave all other options unchanged and click the **OK** button.

The local host will appear in the **SNA Administration** panel, as shown in the figure SNA Administration panel (page 1 on page 381).

### Related Concepts

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

### Related Tasks

“Collect information about the local host” on page 359
“Collect information for your SNA configuration” on page 358.
“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Define links from Communications Server

AIX Use this procedure to define, in the IBM Communications Server for AIX, links (the ″connectivity″) to remote hosts in the SNA network.

It forms part of the overview task to “Configure Communications Server” on page 367 for connection between Component Broker application servers and remote systems over an SNA network.

**Prerequisites:**

- You should have already collected the information about the remote hosts that you need to specify in this procedure. If you have not collected that information, see “Collect Information About Remote Hosts” on page 360.

- This procedure uses the following values as an example:

| *Connection Name* | *Link type* | *Hardware Address* | *CP Name* | *Network Name* | *XID* |
|---|---|---|---|---|---|
| *LINKVTAM* | *IBM Token Ring* | *400012345678* | *MYVTAM* | *MYSNANET* | *04300007* |

  For a description of these values, see "Collect Information About Remote Hosts" on page 360.

- This procedure uses the **xsnaadmin** application provided with Communications Server.

**To configure links to remote hosts in Communications Server, complete the following two stages:**

1. Configure a data link control (DLC) device driver (page 383) to control the use of the network connection to the host on which the Communications Server is running.
2. Configure connections between the local host and remote hosts (page 384).

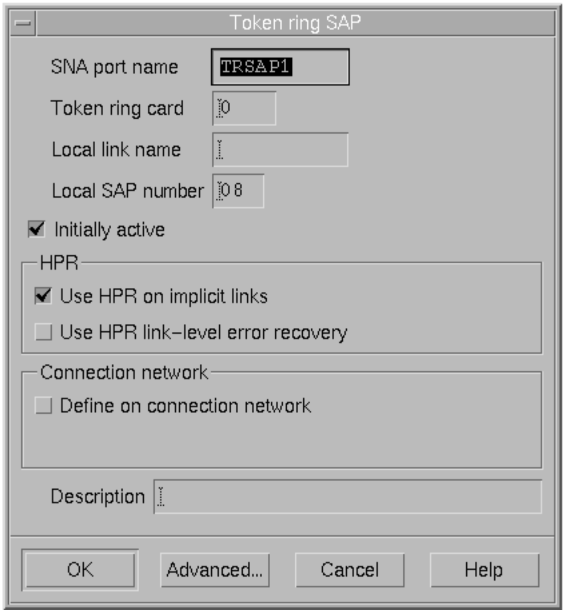
### Configure a data link control (DLC) device driver

To configure a DLC device driver, complete the following steps:

1. Display the **xsnaadmin** application.
2. On the first panel, select **Services - Connectivity - New port**.
3. In the dialog window, select the relevant DLC type you require (for example, select **Token ring card** if your host is using a LAN).

   This displays the panel shown in the figure Token ring SAP panel.

   **Token ring SAP panel**



Token ring SAP

SNA port name: TRSAP1

Token ring card: 0

Local link name:

Local SAP number: 08

☑ Initially active

HPR

☑ Use HPR on implicit links

□ Use HPR link–level error recovery

Connection network

□ Define on connection network

Description:

   If the card number is blank (not 0), you either have not installed a DLC device driver, or you have not rebooted your machine since the installation. You should do so before continuing.

4. Click the **OK** button to add the DLC device.

   The device will appear in the **SNA Administration** panel, as shown in the figure SNA Administration panel (page 1 on page 381).

You can now configure connections between the local host and remote hosts (page 384).

### Configure connections between the local host and remote hosts

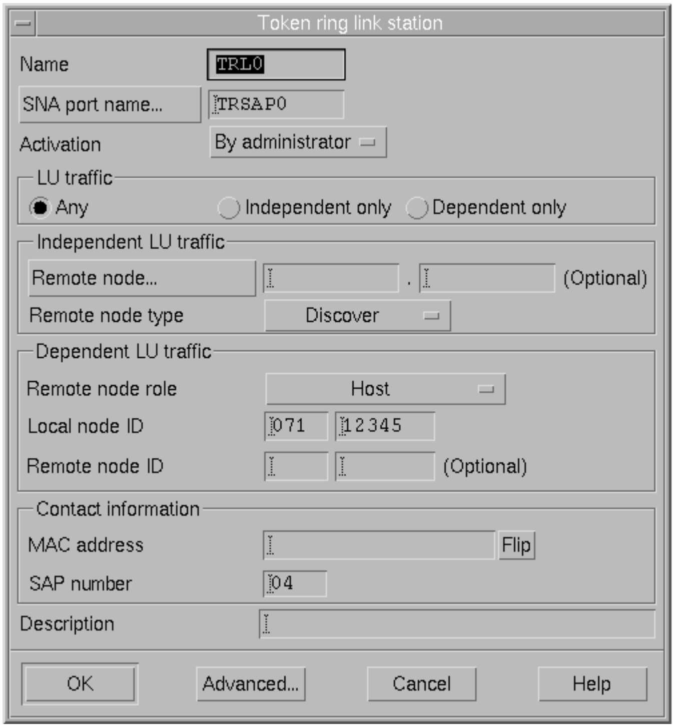
After you have configured a DLC device driver, you can connect your local host to remote hosts by configuring link stations.

To configure a link station to a remote host, complete the following steps:

1. Display the **xsnaadmin** application.
2. Select **Services - Connectivity - New link station**.
3. Select the name of the DLC type you require (for example, select *TRSAP0* if your machine is using a LAN).

   This displays the panel shown in the figure Token ring link station panel (page 384).

   **Token ring link station panel**



Token ring link station

Name: TRL0
SNA port name...: TRSAP0
Activation: By administrator

LU traffic
● Any ○ Independent only ○ Dependent only

Independent LU traffic
Remote node... . (Optional)
Remote node type: Discover

Dependent LU traffic
Remote node role: Host
Local node ID: 071 12345
Remote node ID: (Optional)

Contact information
MAC address:
SAP number: 04

Description:

4. Type the link station name (for example, TRL0).
5. Under Contact Information, type the remote machine address in the MAC address field.

   Click the **OK** button to add the link.

   The link station will appear in the **SNA Administration** panel, as shown in the figure SNA Administration panel (page 1 on page 381).

**Related Concepts**

"IBM Communications Server" on page 34
"Introduction to SNA" on page 471
"Connections to Tier-3 Systems" on page 29

**Related Tasks**

"Collect Information About Remote Hosts" on page 360
"Collect information for your SNA configuration" on page 358.
"Configure Communications Server" on page 367
"Configure an APPC Connection to a Tier-3 System for use by Applications" on page 354

## Define Your own Modegroups in Communications Server

AIX Use this procedure to define, in the IBM Communications Server for AIX, your own modegroups for connections across a SNA network. Normally, you do not need to use this procedure, because Component Broker automatically creates the modegroups it needs, based on information that you define when you "Configure an APPC Connection to a Tier-3 System for use by Applications" on page 354.

You should consider completing this task only if you want your connections to use non-default values for the number of sessions or contentions winners.

This procedure forms part of the overview task to "Configure Communications Server" on page 367 for connection between Component Broker application servers and remote systems over an SNA network.

**Prerequisites:**

- You should have already collected the information about the remote hosts that you need to specify in this procedure. If you have not collected that information, see "Collect information about modenames" on page 365.
- This procedure uses the following values as an example:

| *Modename* | *Max Sessions* | *Min Winners* |
|---|---|---|
| *CB2CICS0* | *10* | *5* |

  The values that Component Broker uses by default are:

| *Modename* | *Max Sessions* | *Min Winners* |
|---|---|---|
| *your_modename* | *20* | *10* |

  Where *your_modename* is the modename that you have chosen for the modegroup used by an APPC Connection.

  For a description of these values, see "Collect information about modenames" on page 365.

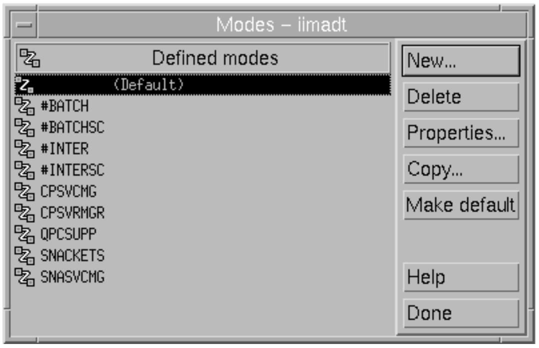
- This procedure uses the **xsnaadmin** application provided with Communications Server.

**To configure your own modegroups (also known as *modes*) in Communications Server, complete the following steps:**

1. Display the **xsnaadmin** application.
2. On the first panel, select **Services - APPC - modes**.

   This displays the panel shown in the figure Modes panel (page 385).

   **Modes panel**

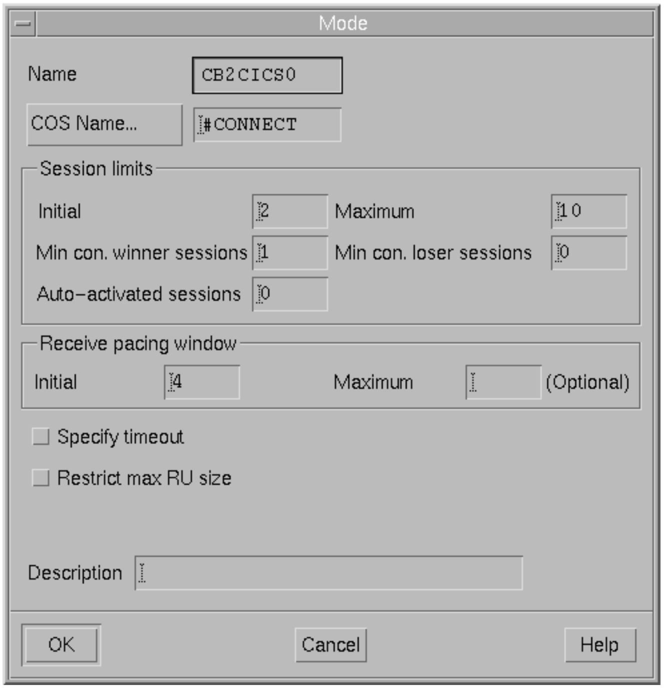


Defined modes

- (Default)
- #BATCH
- #BATCHSC
- #INTER
- #INTERSC
- CPSVCMG
- CPSVRMGR
- QPCSUPP
- SNACKETS
- SNASVCMG

3. Click the **New** button to display the panel shown in the figure Mode panel (page 3).

**Mode panel**



Name: CB2CICS0

COS Name...: #CONNECT

Session limits

| | | | |
|---|---|---|---|
| Initial | 2 | Maximum | 10 |
| Min con. winner sessions | 1 | Min con. loser sessions | 0 |
| Auto–activated sessions | 0 | | |

Receive pacing window

Initial: 4 Maximum: (Optional)

□ Specify timeout

□ Restrict max RU size

Description:

4. Enter the modename (for example, CB2CICS0).
5. Under **Session limits**, type values to define the following session limits:
   - In the **initial** field, type the maximum number of sessions (up to the value of the **maximum** field) that a pair of LUs can have using this mode, unless a different maximum is negotiated using CNOS.
   - In the **maximum** field, type the maximum number of sessions (for each connection).
   - In the **Min con. winner sessions** field, type the minimum number of contention winner sessions (at least 1, and usually about half of the maximum entered above).
   - In the **Min con. loser sessions** field, type the minimum number of contention loser sessions.
   - If you wish to autoactivate some of your sessions on startup, enter the number to be activated in the **Auto-activated sessions** field.
6. Optionally, you can change the performance characteristics of the sessions that belong to the mode. The default values should be sufficient for an initial

configuration. However, better performance can be achieved by matching these values to those defined in the remote LU.

7. After you have entered the values you require for the modegroup, select the **OK** button to add the modegroup to the Communications Server configuration.

The new modename appears in the list of modenames displayed The modename will appear in the **SNA Administration** panel, as shown in the figure SNA Administration panel (page 1 on page 381).

**Related Concepts**

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

**Related Tasks**

“Collect information about modenames” on page 365
“Collect information for your SNA configuration” on page 358.
“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Define Remote Systems to Communications Server

AIX Use this procedure to define, in the IBM Communications Server for AIX, the remote systems that your Component Broker application servers communicate with across an SNA network.

This procedure forms part of the overview task to “Configure Communications Server” on page 367 for connection between Component Broker application servers and remote systems over an SNA network.

**Prerequisites:**

- You should have already collected the information about the remote systems that you need to specify in this procedure. If you have not collected that information, see “Collect Information About Tier-3 Systems” on page 364.
- This procedure uses the following values as an example:

| *LU Name* | *Network Name* | *CP Name of Remote Host* |
|---|---|---|
| *CICSESA* | *MYSNANET* | *MYSNANET.MYVTAM* |

  For a description of these values, see “Collect Information About Tier-3 Systems” on page 364.

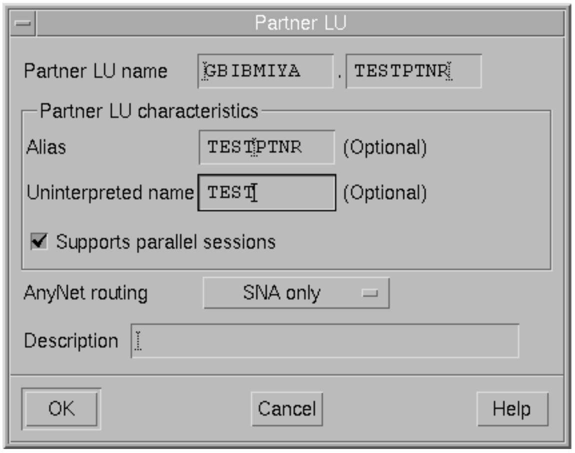
- This procedure uses the **xsnaadmin** application provided with Communications Server.
- If you do not define the partner LU name for the remote system, Component Broker will do this automatically. However, in that case, the CP name will be blank, which is alright if the remote system supports “Introduction to SNA” on page 471. If the remote system does not support APPN, then you should define the partner LU name, se described in this topic.

**To define the remote systems that your application servers communicate with, complete the following steps:**

1. Display the **xsnaadmin** application.
2. On the first panel, select **Services - APPC - New Partner LUs - Remote Node**.

   This displays the panel shown in the figure Partner LU panel (page 388).

   **Partner LU panel**



Partner LU

Partner LU name GBIBMIYA . TESTPTNR

Partner LU characteristics

Alias TESTPTNR (Optional)

Uninterpreted name TEST (Optional)

☑ Supports parallel sessions

AnyNet routing SNA only

Description

3. Enter the fully-qualified partner LU name (MYSNANET.CICSESA).
4. Enter the partner LU alias (CICSESA).
5. Enter the fully qualified CP name of the remote machine.
6. Click the **OK** button, to add the partner LU to the Communications Server configuration.

The new partner LU appears in the **SNA Administration** panel, as shown in the figure SNA Administration panel (page 1 on page 381).

**Related Concepts**

"IBM Communications Server" on page 34
"Introduction to SNA" on page 471
"Connections to Tier-3 Systems" on page 29

**Related Tasks**

"Collect information about the local host" on page 359
"Collect information for your SNA configuration" on page 358.
"Configure Communications Server" on page 367
"Configure an APPC Connection to a Tier-3 System for use by Applications" on page 354

## Configure VTAM with details of your Component Broker APPC Connections

The Virtual Telecommunications Access Method (VTAM) is an IBM product that runs on a mainframe and controls access to mainframe-based systems such as CICS for MVS/ESA, CICS/ESA, CICS/MVS, and CICS/VSE regions. VTAM uses the services of the Network Control Program (NCP) product to connect the mainframe to the network. Therefore, the NCP may need updating so that it can pass requests from a Component Broker host (as a SNA node) to VTAM and the remote system beyond.

VTAM and NCP definitions are coded using macros. This topic shows a few example definitions, which cannot cover the extensive range of network configurations. Therefore, you should use these examples for guidance only; for more information and examples about VTAM and NCP definitions, see the following manuals:

- *VTAM Resource Definition Examples*
- *IBM Network Products Implementation Guide*

**Prerequisites:**

- Before starting this procedure, you should have collected the information about the local host, local LU names, and modegroups that you need, as described in the following topics:
  - "Collect information about the local host" on page 359
  - "Collect information about local application servers" on page 362
  - "Collect information about modenames" on page 365
- You should also have checked that the values you are going to specify to VTAM match the values you specify to Communications Server, as described in "Match Communications Server names with CICS definitions" on page 366.

**To configure VTAM with details of your Component Broker APPC Connections, complete the following two tasks:**

1. "Define your Host and Application Servers to VTAM"
2. "Define Modegroups to VTAM" on page 390

## Define your Host and Application Servers to VTAM

Use the VTAM Physical Unit (PU) macro to define the host computer where the Communications Server (and application server) is running. The example below is a PU macro for a PC, running a Component Broker for Windows NT application server, that is connected to the network with an IBM Token Ring:

```
************************************************************************
*
NT000127 PU    ADDR=C1,            STATION ADDRESS (CAN BE ANY VALUE)  X
               IDBLK=05D,          071 = RS/6000, 05D = OS/2 or NT     X
               IDNUM=98765,        PART OF XID.                        X
               DISCNT=NO,          HANG-UP ON LU LOGOFF                X
               MAXDATA=265,        MAX I-FIELD SIZE                    X
               MAXOUT=7,           RECEIVE PACING WINDOW               X
               MAXPATH=1,          NO OF DIAL-OUT PATHS                X
               MODETAB=MTDFLT,     MODETAB IF LU DOES NOT SPECIFY ONE  X
               SSCPFM=FSS,         LUs NOT SUPPORTING CHAR-CODED MSGS  X
               PACING=0,                                               X
               VPACING=0,                                              X
               PUTYPE=2,                                               X
               ISTATUS=ACTIVE
*
************************************************************************
```

Values coded in the PU definition for the host computer must match the definitions that you create when you "Configure the Local Host in Communications Server" on page 369. For example, VTAM can use either an *Exchange Identifier (XID)* or a *control point (CP) name* to match a request from your host to its PU definition. The

PU definition above has an XID defined. This is made up from the IDBLK and IDNUM values. Therefore Communications Server would be configured with an XID of 05D12345.

The CP name can be coded on a PU definition using the CPNAME parameter. This is not required in the example above because the XID is coded. If you configure both an XID and a CP name in Communications Server, but in the VTAM PU definition you specify only an XID, then you are recommended to make the PU name (NT000127 in this example) the same as the CP name. This will make it easier for you to associate the VTAM PU definition with your host. However if you do use the CPNAME parameter in the PU definition then that CP name must be different from the PU name.

When you are setting up a link between your Component Broker host and VTAM, you must decide which host is to issue the command that establishes the link. One host must *call* and the other must *listen*. It is usual for VTAM to listen and your host to call. However, if you wish to set up VTAM so that it calls your host, VTAM needs to know the address of your host. You define this to VTAM in a **PATH** definition, coded just after the PU definition.

```
***********************************************************************
*
NT01     PATH  GRPNM=WINT,              ECLTYPE=LOGICAL group in NCP  X
               DIALNO=01044000012345678                                X
               GID=1,                                                  X
               PIC=1,                                                  X
               USE=YES
*
***********************************************************************
```

Under the PU definition and, if defined, the PATH macro definitions, are the logical unit (LU) definitions for your Component Broker application servers. The application servers are on the hosts defined in the PU definition, and their LUs are configured automatically in the Communications Server on the same host as the application server.

```
***********************************************************************
*
CBSERV1 LU    LOCADDR=0,ISTATUS=ACTIVE,MODETAB=MTVTAM
*
***********************************************************************
```

The LOCADDR=0 option on the LU definition indicates that the application server’s LU is **independent**. This enables it to communicate with other independent LUs without using VTAM. The MODETAB parameter specifies the name of the VTAM mode table that defined all of the modegroups (modenames) used by the application server.

Examples of mode table entries are shown in “Define Modegroups to VTAM”.

## Define Modegroups to VTAM

Use the VTAM MODETAB macro to create a VTAM mode table to define the modegroups used for communication between by your application server and a mainframe-based CICS region.

The example macro below shows part of a VTAM mode table called MTVTAM, which defines a number of modegroups that include the entry for modename CB2CICS0. The mode table used by a CICS region must have a definition for all of the modegroups (modenames) it uses and a definition for the SNASVCMG mode group. This modegroup is used by Communications Server for network management requests.

The values that you code in the MODEENT definition must match the modegroups created for your application server in Communications Server. Normally, the modegroups for your application servers are created automatically when you activate the system management Configuration that defines those servers. Alternatively, you can create your own modegroups for application servers, as described in “Define Your own Modegroups in Communications Server” on page 375.

```
MTVTAM   MODETAB
***********************************************************************
* MODE TABLE FOR CICS                                                 *
***********************************************************************
          :            :            :            :          :
*
* Modename CB2CICS0 - Parallel_Sessions=yes
*
CB2CICS0 MODEENT LOGMODE=CB2CICS0,                                     X
               TYPE=0,             ONLY TYPE RECOGNISED                X
               FMPROF=X'13',       SNA                                 X
               TSPROF=X'07',       SNA                                 X
               PRIPROT=X'B0',      PRIMARY PROTOCOL                    X
               SECPROT=X'B0',      SECONDARY PROTOCOL                  X
               COMPROT=X'79A5',    COMMON PROTOCOL                     X
               SSNDPAC=X'00',                                          X
               SRCVPAC=X'00',                                          X
               RUSIZES=X'8989',    RUSIZES IN-4096 OUT-4096            X
               PSNDPAC=X'00',                                          X
               PSERVIC=X'060200000000000000122F00'
*
* Modename SNASVCMG - required for parallel sessions
*
SNASVCMG MODEENT LOGMODE=SNASVCMG
          :            :            :            :          :
         MODEEND
***********************************************************************
* END OF MODE TABLE FOR CICS                                          *
***********************************************************************
```

### Related Concepts

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

### Related Tasks

“Collect information for your SNA configuration” on page 358.
“Match Communications Server names with CICS definitions” on page 366.
“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Configure a Server for Connections to tier-3 Systems

Use this procedure to configure a server or server group with server-wide details for connections to one or more tier-3 Systems; for example, a CICS regions and IMS servers.

**Prerequisites:**

The maximum limits defined for a server in this task specify server-wide limits extra to those defined for individual applications by their Connection objects.

- When you configure a Connection object, you set the limits for applications that use that Connection to the tier-3 system it identifies.
- In this task, you define the limits for all connections of the same type ECI or HOD that are used on a server to connect to tier-3 systems.

If you use this task to configure a server group, the same limits are applied to each server that is a member of the server group.

**To configure the Connection details for an application server, complete the following steps:**

1. Expand the Configuration that contains the Server (freestanding) or Server Group.
2. Expand the **Servers (freestanding)** folder or **Server Groups** folder.
3. Edit the Server (freestanding) or Server Group, to set server-wide characteristics for ECI connections to tier-3 systems. To do this, complete the following steps:
   a. From the pop-up menu of the Server (freestanding) or Server Group, click **Edit**. This displays the Object Editor.
   b. Click the **Application Adaptors** notebook tab, then set the following attributes to your chosen values:

      **maximum number of ECI connections**
      The maximum number of connections allowed at the same time for all ECI Connections from this server; default, 12.

      **maximum wait time for ECI connections**
      The maximum time that a connection request is delayed if the maximum number of ECI connections from this server already exists; default, 300.

      **maximum number of HOD connections**
      The maximum number of connections allowed at the same time for all HOD Connections from this server; default, 12.

      **maximum wait time for HOD connections**
      The maximum time that a connection request is delayed if the maximum number of HOD connections from this server already exists; default, 300.

      **maximum number of Generic connections**
      The maximum number of connection instances allowed for Generic Connections from this server; default 12.

      **maximum wait time for Generic connections**
      The maximum time that a connection request is delayed if the maximum number of Generic connections from this server already exists; default, 300.

**Notes:** If the **maximum number of ... connections** limit is reached, a connection request is delayed for the period specified by the corresponding **maximum wait time for ... connections** attribute. A value of 0 (zero) implies no maximum limit.

The **maximum number of connections** and **maximum wait time** attributes of a corresponding Connection configured for an Application can further limit the connections for that application. If the Connection's **maximum number of connections** limit is reached before the server-wide limit, a connection request is delayed for the period specified by the Connecion's **maximum wait time** attribute.

c. If your application server is using security services, you need to specify the userid and password that the application server will use to connect to the tier-3 CICS region.

   To do this, click the **Security Service** notebook tab, then set the following attributes to your chosen values:

   **data system principal**
      The user ID that the server will use when connecting to tier-3 systems

   **data system password**
      The password that the server will use when connecting to tier-3 systems

   **security enabled**
      Check that this is set to **yes**.

   **Note:** You can define only one userid and password for a server group or freestanding server. This has the following effects:

   - All servers in a server group use the same userid and password
   - If a server needs to connect to more than one tier-3 system (for example, a CICS region and an IMS system) then all tier-3 systems must use the same userid and password for the application server.

d. When you have finished, click the **OK** button to apply your changes and close the Object Editor window.

4. To verify that your application Configuration is valid, and to apply the changes to the runtime configuration of your application servers, activate the Configuration. To do this, from the pop-up menu of the Configuration click **Activate**.

### Related Concepts

"Connections to Tier-3 Systems" on page 29

### Related Tasks

"Configure a new HOD Connection to a Tier-3 System" on page 344
"Configure a new ECI Connection to a Tier-3 System" on page 348
"Configure a new APPC Connection to a Tier-3 System" on page 353
Configure Server Security Information for Connection to Tier-3 Systems
"Configure the iPAAServices application onto the application server" on page 394

# Configure the iPAAServices application onto the application server

Use this procedure to configure the iPAAServices application onto an application server. The iPAAServices application provides the Procedural application adaptor (PAA) services needed to support connections to tier-3 systems. Those services can then be used by one or more applications running on Component Broker application servers.

The default configuration for each application server is set so that it can use the basic set of Component Broker services for the server to be able to run. By default, application servers do not have the PAA services to support connections to tier-3 systems.

When you install the CICS and IMS application adaptor run time, the installation process automatically installs the **iPAAServices** application needed to provide the PAA services, and adds it to the folder of available applications for the System Manager.

An application that wants to use these services does not need to have the service configured onto its application server. You can configure the service onto a server that you want to provide these services, and an application in any server can make use of these services.

**To configure the iPAAServices application onto an application server, complete the following steps:**

1. Add the iPAAServices application to the Configuration that contains your ECI Connection. If you have already added iPAAServices to your Configuration, you do not need to do so again; you can use the existing iPAAServices Application.

   To add iPAAServices to your Configuration, complete the following steps:

   a. Expand the Available Applications folder (at the top of the tree in the System Manager window)

   b. On the pop-up menu of the iPAAServices Application, click **Drag**.

   c. On the pop-up menu of your Configuration, click **Add Application**

   The **Add Application** action creates an iPAAServices Application within your Configuration.
2. Configure the IPAAServices Application onto the application server that is to provide those services. To do this, complete the following steps:

   a. On the pop-up menu of the iPAAServices Application (within your Configuration), click **Drag**

   b. Expand the Server Groups folder or Server (freestanding) folder that contains a definition for the application server.

   c. On the pop-up menu of the Server Group or Server (freestanding) that is to provide the service, click **Configure Application**

   **Note:** If you want several application servers to provide the PAA services, configure the iPAAServices Application on all those servers (or their Server Group).

### Related Concepts

"CICS Transaction Gateway" on page 34
"Connections to Tier-3 Systems" on page 29

**Related Tasks**

Configure a new HOD Connection to a Tier-3 System
Configure a new APPC Connection to a Tier-3 System
Configure Server Security Information for Connection to Tier-3 Systems

# Chapter 14. Administer Component Broker Services

This topic provides information about administering Component Broker services for use in an existing application environment.

For example, the default configuration for each application server is set so that it can use the basic set of Component Broker services for the server to be able to run. By default, application servers do not have the functions (for example, homes) to support the creation of lifecycle, event, or notification objects. You can use the procedure “Configure a Server to Provide Extended Component Broker Services” on page 406 to add the service support needed for those objects.

**Related Concepts**

“The Cache Service”

**Related Tasks**

“Chapter 5. Configure a new Application Environment” on page 133
“Configure a Server to use the Query Service” on page 402
“Configure a Server Group to use the Transaction Service” on page 403
“Configure a Freestanding Application Server to use the Transaction Service” on page 405
“Configure a Server to Provide Extended Component Broker Services” on page 406
“Change the userid and password used to access a DB2 Database” on page 243
“Configure an Application to use the Cache Service” on page 408
“Configure an Event Channel for use by Applications” on page 410
“Configure a Client Style to use Session Services” on page 411
“Configure a Client Style to use a Specific Factory Finder” on page 412

## The Cache Service

The *cache service* is used to improve performance and concurrency for applications reading data from relational databases.

You can control the characteristics of the cache service by the following attributes of its Profile Class object.

- Lock confidence (pessimistic or optimistic)
- Refresh interval (specified in number of seconds to retain data in memory)
- Defer update (yes or no)
- Access (read, write, upgrade; applicable to DB2 only)

Each managed object type (interface name) can have its own Profile Class to set its own cache characteristics.

**Pessimistic and Optimisitic caching:**

The cache service has the following caching options:

**pessimistic caching**

This option causes data to be read from the database for each transaction and locked in the database while data is cached in memory. Locks are released at the end of the transaction. Data is retained in memory after commit. **Pessimistic caching should be used for data that changes frequently and where the application requires the most current copy of the data.**

Data is normally read with a row share lock maintained on the database record to prevent concurrent updates. This happens if the **access** attribute of the Profile Class is set to **read** or **upgrade**. If the **access** attribute of the Profile Class is set to **write**, then data is read with a row exclusive lock maintained on the database record to serialize all access. Obtaining an exclusive lock (when access is set to **write**) is useful when your application has to read and update a sequential counter to serialize concurrent transactions that attempt to read and the update the counter.

**Note:** Pessimistic caching is the default for the cache service.

**optimistic caching**

In optimistic caching, the data is read from the database, but the database lock is immediately released after the read. The data is cached in memory and reused across transactions. A request for data is satisfied from the cache before reading the database. A refresh interval is specified to control the amount of time where the cached data in memory can become out of sync with the database. (The data is refreshed according to the refresh interval.) The size of the cache can also be specified by the **cache size** attribute of the iDefault Profile used by an application. The cache service uses an LRU algorithm to remove unneeded entries from the cache (to manage the cache to this specified size). **Optimistic caching should be used for applications that do not need the most current data and for data that is mostly read-only and does not change often.**

If data that is being cached optimistically is updated, the record in the database may have been updated by another transaction or may have been deleted. When optimistically cached data is written back to the database, a read and compare is done to make sure that the data record still exists in the database and the data values have not changed. This prevents any *lost updates* from occurring. Only the attributes that are actually updated are compared. Strings longer than 256 bytes *are not compared* on update.

The **access** attribute of the Profile Class has no effect for optimistic caching (if the **lock confidence** attribute is set to **optimistic**).

**Data caching for server groups:**

When using server groups, each server within the server group has a separate data cache. It is possible for different clients to see different values for the same object. The caches refresh themselves from the database after a period of time specified by the cache refresh interval and eventually all caches contain the current value.

To minimize this problem, set the refresh interval to a small time interval or use *pessimistic* caching. Setting the refresh interval to a small value or using *pessimistic* causes more frequent access to the database.

**What the Cache Service Does:**

To improve performance and concurrency for applications, the cache service does the following:

1. Keep read only data (or mostly read only data) resident in memory. If data does not change very often it should not be necessary to have to reread the data from the database on every transaction. This is called **optimistic caching**.

2. Data whose application must have the most current value must be read from the database every transaction.
3. Data whose access and update must be serialized (for example, counter or sequential number generator stored in a database) is accessed with an exclusive lock if the **access** attribute of the Profile Class is set to **write**. Other data can be accessed in shared mode to allow concurrent read activity, if the **access** attribute of the Profile Class is set to **read** or **upgrade**.
4. For DB2 databases, allow users to specify whether to use cursor stability (CS) locking mode or (RS) read stability locking mode at the database. Use of CS locks improves concurrency at the database server but does not guarantee transaction serialization. RS locks guaranteee transaction serialization but usually at some cost of concurrency and performance. You are recommended to use CS locking. This locking option is set by the **lock confidence** attribute of the Profile Class. A lock confidence of **optimistic** uses CS locking; a **lock confidence** of **pessimistic** uses RR or RS locking.
5. Allow the user to specify whether to defer updates until commit. If a transaction updates two attributes on an object instance, using deferred updates results in only one SQL update. If the updates are done immediately when the attribute value changes, then two SQL updates are made. The advantage of deferring updates is reduced SQL calls and better performance. The advantage of not deferring updates is that the application gets immediate feedback (via an exception) if the update violates some database constraint. If you defer updates until commit and then some database constraint is violated, the only feedback that the client application receives is that the transaction commit failed and the transaction is rolled back. In this case, the client application has to repeat the entire transaction. Even so, it is usually recommended to defer updates.

The cache service maintains a private cache for each transaction. If two transaction are reading the same record from the database at the same time, there are 2 copies of the record in memory.

**Configuring Characteristics of the Cache Service:**

You configure the characteristics of the cache service by the following attributes of a Profile Class object. Each application can have its own Profile Class (and therefore its own characteristics for the cache service).

- **access** (read, upgrade, or write)
- **defer update** (yes or no)
- **lock confidence** (pessimistic or optimtistic)
- **refresh interval** (the number of seconds to retain data in memory)

An application developer can set the characteristics of the cache service for an application within its application family package. To do this, the application developer creates a Profile Class Install object and related objects within the application family’s DDL file. Then when the application has been installed into Component Broker, and the system administrator has configured the application onto a server, the characteristics of the cache service are set automatically.

A system administrator can set the characteristics of the cache service for an application through the System Manager user interface. This would be needed only if an application is to use the cache service and the characteristics had not already been set within the application family package, or to change the characteristics.

How to configure the characteristics of the cache service using the System Manager user interface is described in the topic “Configure an Application to use the Cache Service” on page 408.

Valid cache options are given in the following tables:

- **Cache Options for Optimistic Caching** (page 400)
- **Cache Options for Pessimistic Caching** (page 401)

### Cache Options for Optimistic Caching

| Defer Updates | Refresh Interval | Access | Comments |
|---|---|---|---|
| yes | =0 | not used | **Option o1**. Database locks are released after reading the data. This minimizes database lock resource usage. Updates are deferred. Because the *refresh interval* is zero, data is purged from memory at the end of the transaction. Every transaction rereads data from the database so data is current. **This is the recommended option for long running transactions. It prevents the long running transactions from causing concurrency problems at the database server.** |
| yes | >0 | not used | **Option o2**. Same as **Option o1**, except data stays in memory at the end of the transaction. The *refresh interval* specifies how many seconds data stays in memory until a refresh. **This is the recommended option for read only data like TAX tables, ZIP CODE tables and product descriptions.** |
| no | =0 | not used | **Option o3**. Similar to **Option o1** except updates are not deferred. An SQL update is executed whenever an attribute changes value. **If the transaction needs immediate feedback on any database integrity violations, use this option.** |

| Defer Updates | Refresh Interval | Access | Comments |
|---|---|---|---|
| no | >0 | not used | **Option o4**. Similar to **Option o2** except data stays in memory and can be reused by other transactions. Updates are not deferred. The *refresh interval* specifies the number of seconds for data to stay in memory. |

**Note:** The **access** attribute of the Profile Class has no effect for optimistic caching (if the **lock confidence** attribute is set to **optimistic**).

### Cache Options for Pessimistic Caching

| Defer Updates | Refresh Interval | Access | Comments |
|---|---|---|---|
| yes | not used | read or upgrade | **Option p1**. Data is locked (share locks) in the database during the transaction. Data is purged from cache at commit. Updates are deferred until commit. **This is the recommended option for data that changes often or if the application must have current data. For example, inventory quantities or bank balances.** |
| no | not used | read or upgrade | **Option p2**. Same as **Option p1**, except updates are not deferred. An SQL update is performed whenever as object attribute changes value. |
| yes | not used | write | **Option p3**. Data is locked exclusively in the database during the transaction. Data is purged from cache at commit. Updates are deferred until commit. This is the recommended option for data that must be serialized for both reading and writing and there is a high probability of contention among multiple transactions such as reading and updating a sequential number generator. |

| Defer Updates | Refresh Interval | Access | Comments |
|---|---|---|---|
| no | not used | write | **Option p4**. Same as **Option p3**, except updates are not deferred. An SQL update is performed whenever an object attribute changes value. |

**Note:** The **refresh interval** attribute of the Profile Class has no effect for pessimistic caching (if the **lock confidence** attribute is set to **pessimistic**).

### Related Tasks

"Configure an Application to use the Cache Service" on page 408
"Configure DB2 to use the CBConnector Data Cache Facility" on page 470

## Configure a Server to use the Query Service

Use this procedure to *configure* an application server to use the Query Service. You need to do this if any user applications that are to run on the server need to use the Query Service.

**To *configure* an application server to use the Query Service, complete the follow steps before starting the server:**

1. Create your database (*your_database_name*) and appropriate tables
2. Bind all *PO.bnd files to your database; for example, by using the following commands:
   ```
   idatapre xyzPO.bnd
   <your_database_name> bind
   idatapre abcPO.bnd <your_database_name> bind
   ```
3. Bind the following files from your Component Broker **\etc** directory:
   ```
   idatapre db2cntcs.bnd <your_database_name> bind
   idatapre db2cntrr.bnd <your_database_name> bind
   ```
4. On the Host Image that represents the host computer on which the application server is to run, use the **Load DDL** action to load the following DDL files:
   - The iDB2IMServices provided by Component Broker
   - Both the basic and specific DDL files for your application
5. Copy the DLL files for the user application to the Component Broker **\bin** directory on the host computer on which the application server is to run. If you have configured your application onto the server correctly within a system management Configuration then, when you activate that Configuration, the DLL files are downloaded automatically to the host computer on which the server is to run.

### Related Tasks

"Add an Application into a Configuration of your Application Environment" on page 228
"Configure an Application onto a Freestanding Server" on page 186
"Activate a Configuration" on page 256

## Configure a Server Group to use the Transaction Service

By default, the transaction service is enabled for a server group, but with default attribute values.

If you wish to change the default attribute settings for a freestanding server, see “Configure a Freestanding Application Server to use the Transaction Service” on page 405.

If you wish to change the default attribute settings for a server group, you should use the System Manager user interface to complete the following steps.

1. Expand the Management Zone that defines your application environment
2. Expand the Configurations folder
3. Expand the Configuration within which the server group is defined
4. Expand the Server Groups folder
5. If you want to change the default attributes that apply to all servers in the server group, or want to change the log directory for the SGCP and SGGW servers of a controlled server group, complete this step.

   If you only want to change the log directory of an application server that is a member of the server group, click here to go to the appropriate step.

   a. From the pop-up menu of the Server Group model, click **Edit**. This displays the Object Editor window for the Server Group model.

   b. In the Object Editor Notebook, click the **Transaction Service** tab.

      You can change the following attributes to control the transaction service for all servers that are members of the server group:

      **retry restricted**
      Whether or not to restrict the number of times that the transaction service retries a commit, rollback, or forget operation if a problem such as a communications failure occurs. Set this to `yes` if you want the Transaction Service to retry a failed operation until it completes successfully or until the `commit retry limit` is exceeded. If set to `no` (the default), the Transaction Service retries a failed operation until the operation completes successfully.

      **commit retry limit**
      The number of times that the server should retry a failed attempt to commit or rollback data updates; default, 0. This value is only applicable if the **retry restricted** attribute is set to `yes`.

      **heuristic direction**
      What the server should do with uncommitted data changes if a failure occurs; default, `rollback`. For `rollback`, the Transaction Service rolls back all uncommitted data changes (at the time of failure) to their last committed state. For `commit`, the Transaction Service commits all uncommitted data changes at the time of failure. rollback

      **log file size**
      The size, in bytes, of the transaction-log-extent files that the transaction service uses to log transactional state information. The value can be any multiple of 65536 (64KB) in the range 65536 through 1048576 (1MB); default 65536 bytes. If you type any other value, it is rounded up to the next multiple of 64KB.

For a controlled server group you can change the following attribute to control the transaction service log directory for the SGCP and SGGW servers:

**log directory**
: The directory in which the transaction recovery data is stored for the SGCP and SGGW servers. This must be the name of a directory that already exists on the managing host for the controlled server group, as detailed in the note below (page 404).

c. To save the changes and close the Object Editor Notebook, click the **OK** button.

6. If you want to change the log directory of an application server that is a member of the server group, complete the following steps:
   a. Expand the Server (member of groups) folder
   b. From the pop-up menu of the Server (member of group) model, click **Edit**. This displays the Object Editor window for the application server.
   c. In the Object Editor Notebook, select the **Transaction Service** tab.

      Specify the directory to be used for the transaction service log files on the following attribute:

      **log directory**
      : This must be the name of a directory that already exists on the host on which the server is to be used, as detailed in the note below (page 404).

      **Note:** The values of other attributes on the Transaction Service notebook page are set on the Server Group model, as described in an earlier step of this procedure (page 403).
   d. To save the changes and close the Object Editor Notebook, click the **OK** button.

   Repeat this step for each Server (member of group) model in the server group.
7. The changes will be applied to the server group in your enterprise when you next activate the Configuration containing the Server Group model.

**Note:** The transaction service will not be available in a server if the **log directory** attribute specifes a directory that does not exist. This is because the transaction service creates log files in this directory that are required to preserve transaction integrity in the event of server failure. You must therefore check that the directory specified is valid. If the directory does not exist the transaction service will fail to initialize and it will not be possible to begin a transaction on that server.

### Related Concepts

"The Transaction Service Log" on page 418

### Related Tasks

"Configure a Controlled Server Group" on page 339
"Configure a Freestanding Application Server to use the Transaction Service" on page 405
"Configure a Server to Provide Extended Component Broker Services" on page 406

## Configure a Freestanding Application Server to use the Transaction Service

The default configuration for each application server is set so that the transaction service is enabled, but with default attribute values.

If you want to change the transaction service attribute values for an application server that is a member of a server group, or want to change the values for a server group, see “Configure a Server Group to use the Transaction Service” on page 403.

If you want to change the default attribute values for a freestanding server, you should use the System Manager user interface to complete the following steps:

1. Expand the Management Zone that defines your application environment
2. Expand the Configurations folder
3. Expand the Configuration within which the application server is defined
4. Expand the Servers (freestanding) folder
5. From the pop-up menu of the Server (freestanding) model, click **Edit**. This displays the Object Editor window for the Server model.
6. In the Object Editor Notebook, select the **Transaction Service** tab. The following attributes relate to the transaction service:

   **retry restricted**
   Whether or not to restrict the number of times that the transaction service retries a commit, rollback, or forget operation if a problem such as a communications failure occurs. Set this to `yes` if you want the Transaction Service to retry a failed operation until it completes successfully or until the `commit retry limit` is exceeded. If set to `no` (the default), the Transaction Service retries a failed operation until the operation completes successfully.

   **commit retry limit**
   The number of times that the server should retry a failed attempt to commit or rollback data updates; default, 0. This value is only applicable if the **retry restricted** attribute is set to `yes`.

   **heuristic direction**
   What the server should do with uncommitted data changes if a failure occurs; default, `rollback`. For `rollback`, the Transaction Service rolls back all uncommitted data changes (at the time of failure) to their last committed state. For `commit`, the Transaction Service commits all uncommitted data changes at the time of failure. rollback

   **log directory**
   The directory in which the transaction recovery data is stored for this server. This must be the name of a directory that already exists on the host on which the server is to be used, as detailed in the note below (page 406).

   **log file size**
   The size, in bytes, of the transaction-log-extent files that the transaction service uses to log transactional state information. The value can be any multiple of 65536 (64KB) in the range 65536 through 1048576 (1MB); default 65536 bytes. If you type any other value, it is rounded up to the next multiple of 64KB.
7. To save the changes and close the Object Editor, click the **OK** button.

8. The changes will be applied to the server in your enterprise when you next activate the Configuration containing the Server (freestanding).

**Note:** The transaction service will not be available in an application server if the **log directory** attribute specifes a directory that does not exist. This is because the transaction service creates log files in this directory that are required to preserve transaction integrity in the event of server failure. You must therefore check that the directory specified is valid. If the directory does not exist the transaction service will fail to initialize and it will not be possible to begin a transaction on that server.

**Related Concepts**

"The Transaction Service Log" on page 418

**Related Tasks**

"Configure a Server to Provide Extended Component Broker Services"
"Configure a Server Group to use the Transaction Service" on page 403

## Configure a Server to Provide Extended Component Broker Services

The default configuration for each application server is set so that it can use the basic set of Component Broker services for the server to be able to run. By default, application servers do not have the functions (for example, homes) to support the creation of lifecycle, event, or notification objects. You can use this procedure to add the service support needed for those objects.

The Component Broker installation process automatically installs the applications for the lifecycle, event, and notification services, and adds those applications to the folder of available applications for the System Manager.

An application that wants to use any of these services does not need to have the service configured onto its application server. You can configure the service onto a server that you want to provide these services, and an application in any server can make use of these services.

**To configure a server to provide any of these services, complete the following steps using the System Manager user interface:**

1. Create an Application model in your Configuration for each service application that a server in that Configuration may need to use. To do this, complete the following steps:
   a. Display the home of the system manager network, and ensure that the user-level is set to **Expert**
   b. Expand the Application Families folder, to display the following service application families and their applications:

| Application Family | Application |
|---|---|
| iEventServices | iEventService |
| iEventServices | iDefaultCellEventChannel |
| iLifeCycleServices | iLifeCycleService |
| iNotificationService | iNotificationService<br>iDefaultCellNotifyChannel |

**Note:** The Component Broker cell has only one default event channel and one default notification channel, provided by the iDefaultCellEventChannel and iDefaultCellNotifyChannel applications respectively. These applications should be configured onto only one server.

c. On the pop-up menu of the service application in the Available Applications folder, click **Drag**.

d. On the pop-up menu of your Configuration, click **Add Application**

The **Add Application** action creates an Application model, within your Configuration, for each of the service applications that you chose to define. For more information about adding applications to Configurations, see “Add an Application into a Configuration of your Application Environment” on page 228.

2. Configure one or more of the service applications onto the server that is to provide those services. To do this, complete the following steps:

   a. Expand your Configuration (that now defines the service applications and the server that is to provide those services)

   b. Expand the Applications folder

   c. On the pop-up menu of the service Application model, click **Drag**

   d. Expand the Server Groups folder or Server (freestanding) folder

   e. On the pop-up menu of the Server Group or Server (freestanding) that is to provide the service, click **Configure Application**

   **Notes:**

   - If you want several application servers to provide services, configure the service application on all those servers (or their Server Group model).
   - If you want to use the default event channel, configure the iDefaultCellEventChannel application onto only one of the servers that provides the event service (iEventService). Likewise, if you want to use the default notification channel, configure the iDefaultCellNotifyChannel application onto only one of the servers that provides the notification service (iNotificationService).
   - If your server group contains more than one server, do not configure either the iDefaultCellEventChannel application or the iDefaultCellNotifyChannel application onto the Server Group model. Instead, you can define a Server (freestanding) to provide one or both of the default event channel and default notification channel within your Configuration.
   - If you do not want to use the default event channel, you do not need to configure the iDefaultCellEventChannel application onto a server in your Configuration. You can define your own Event Channel to match your needs and configure that onto your Server Group or Server (freestanding).
   - If you do not want to use the default notification channel, you do not need to configure the iDefaultCellNotifyChannel application onto a server in your Configuration. You can define your own Notification Channel to match your needs and configure that onto your Server Group or Server (freestanding).

When the Configuration is next activated, the services will be provided by the servers onto which you have configured the service applications.

### Related Tasks

Install an Application using the Load Application Action
“Add an Application into a Configuration of your Application Environment” on page 228

“Configure an Application onto a Freestanding Server” on page 186
“Configure a Freestanding Application Server to use the Transaction Service” on page 405

## Configure an Application to use the Cache Service

Use this procedure to *configure* an application to use the data cache service to improve its performance in reading data from a relational database. This determines the amount of memory allocated for the cache and other cache options that apply to particular managed object classes.

**Prerequisites:**

You only need to complete this task if an application is to use the cache service and the characteristics have not already been set within the application family package.

This task involves creating the following system management objects, editing some of their attributes, and creating relationships between them:

- A Profile model called **iDefault**, to set the default cache size
- One or more Profile Class models, to set different cache characteristics for one or more managed object classes. Note that several managed object classes that need the same cache characteristics can use the same profile class.

You also relate these objects to the Application model and its Managed Object Class models that are to use the data cache.

The application should already have been “Add an Application into a Configuration of your Application Environment” on page 228.

If the Configuration does not contain models for the Managed Object Classes that are to use the data cache, you must create new Managed Object Classes and configure them onto the Application model. The Managed Object Class models must have the same names as their corresponding Managed Object Class Install objects. To display the Managed Object Class Install objects, select the **Expert** user-level, then expand the Available Applications folder, the Application object, and its *Provided Managed Object Classes* relationship.

**To configure the data cache options for an application, complete the following steps:**

1. Expand the Configuration that contains models for your Application and the Servers that it is to run on.
2. Create a new Profile model. It must have a name of **iDefault**. If the Profile model already exists, omit this step.
3. Edit the **iDefault** Profile model to set the **cache size** attribute to the default number of 1K blocks of memory to be allocated for all data caches. (After activating the Configuration (later), you can change the runtime value for individual data caches on running servers, by editing the Profile Image in the Server Image.)
4. Create one or more Profile Class models. Each Profile Class model defines different characteristics for the data cache, for use by different managed object classes. You can choose any unique name for each Profile Class model. Note that several managed object classes that need the same cache characteristics can use the same profile class.

5. Edit each Profile Class model, to set the following attributes to your chosen values:

   **access**
   Normally you can set this attribute as **read** (the default). However, to obtain an exclusive lock on data in the database, set this value to **write** and set **lock confidence** to **pessimistic**.

   **defer updates**
   You are recommended to set this attribute to **yes**.

   **lock confidence**
   For data that is relatively stable and does not change frequently, set this attribute to **optimistic**. For data that changes frequently and where it is important that your applications have the most current data at all times, set this attribute to **pessimistic**.

   **refresh interval**
   Set this attribute to the number of seconds that the cache retains data before rereading it from the database. If **lock confidence** is set to **pessimistic**, **refresh interval** is not used.

   For more information about valid data cache options, see the following tables:

   - **Cache Options for Optimistic Caching** (page 400)
   - **Cache Options for Pessimistic Caching** (page 401)

   Relate the **iDefault** Profile to the Profile Classes and Application that are to use it, by completing steps 6 through 8:

6. On the pop-up menu of the **iDefault** Profile model, click **Drag**.
7. On the pop-up menu of each Profile Class model created in step 4, click **Configure Profile**. This creates a *Configured Profile Classes* relationship between the Profile model and the Profile Class model. You can see the relationships created by expanding the Configured Profile Classes folder within the Profile. (If selecting several Profile Class models at the same time, you must display them in List View and use the **Selected** menu bar choice to select actions.)
8. On the pop-up menu of the Application model, click **Configure Profile**. This creates *Provided Profiles* relationships between the Profile model and the Application model. You can see the relationships created by expanding the Provided Profiles folder within the Application.

   Relate each Profile Class to the Managed Object Classes and the Application that are to use the cache configuration, by completing steps 9 through 11: (If the Configuration does not contain models for the Managed Object Classes that are to use the data cache, see prerequisites (page 408).)

9. On the pop-up menu of a Profile Class model (created in step 4), click **Drag**.
10. On the pop-up menu of each Managed Object Class model that is to use the Profile Class model, click **Configure Profile Class**. This creates the *Configured Managed Object Classes* relationship between the Managed Object Class and the one Profile Class that it is to use. (If selecting several Managed Object Class models at the same time, you must display them in List View and use the **Selected** menu bar choice to select actions.)

    **Note:** Each Managed Object Class should use only one Profile Class. If you later want to change cache options for a Managed Object Class, just edit its Profile Class and reactivate the Configuration.

11. On the pop-up menu of the Application model, click **Configure Profile Class**. This creates *Provided Profile Classes* relationships between the Profile model and the Application model. You can see the relationships created by expanding the Provided Profile Classes folder within the Application.
12. Repeat steps 9 to 11 for each other Profile Class model.

When the Configuration is next activated, a data cache with the characteristics defined in this procedure will be set up on each server that the Application model is configured on.

**Related Concepts**

"The Cache Service" on page 397

**Related Tasks**

"Configure DB2 to use the CBConnector Data Cache Facility" on page 470

## Configure an Event Channel for use by Applications

Use this procedure to *configure* a new event channel for use by one or more applications.

**Prerequisites:**

You only need to complete this task if an application is to use the new event channel and the characteristics have not already been set within the application family package.

An application that needs to use the event channel should already have been "Add an Application into a Configuration of your Application Environment" on page 228.

**To configure a new event channel for use by applications, complete the following steps:**

1. Expand the Configuration that contains models for your Application and the Servers that it is to run on.
2. Create a new Event Channel. To do this, complete the following steps:
   a. From the pop-up menu of the Configuration click **New - Event Channel**

      This displays a dialog window for you to specify the name of the new Event Channel. This name is used for system management only and does not need to be the same as the name of the event channel in the system name space.
   b. Specify a name, unique within the Configuration. If you specify a name that is already in use, you will be prompted to type a new name.
   c. To create the Event Channel, click the **OK** button.
3. Edit the Event Channel, to set its attributes to your chosen values, by completing the following steps:
   a. Expand the Event Channels folder. You should see an object for the new Event Channel that you have just defined. If not, refresh the Information Controller window, then if needed repeat step .
   b. On the pop-up menu of the Event Channel, click **Edit**. This displays the Object Editor window.
   c. Set the following attributes to your chosen values:

**event delivery retry count**
The number of times that the event channel should try to send an event notification to an event consumer after an initial failure to send the event. Default: 5

**relative name**
The relative name below *host/resources/event-channels*.

The event channel may also bind the relative name in the work group and/or cell branch of the name tree, depending upon the values set for the visible in xxxxxx name tree attributes. For example, if you set **visible in work group name tree** set to **yes**, the event channel is bound, using the relative name, below *work group/resources/event-channels*.

**visible in cell name tree**
Whether or not the event channel can be found in the cell name tree. Default: no

**visible in host name tree**
Whether or not the event channel can be found in the host name tree. Default: no

**visible in work group name tree**
Whether or not the event channel can be found in the work group name tree. Default: no

Optionally, you can also specify a **description** of the Event Channel, as a text string up to 256 ASCII characters.

d. To apply any changes to the attributes, and close the Object Editor window, click the **OK** button.

4. Relate the Event Channel to each Application that is to use it, by completing the following steps:
   a. On the pop-up menu of the Event Channel, click **Drag**.
   b. Expand the Applications folder, to display the applications that are to use the event channel.
   c. On the pop-up menu of the Application, click **Configure Event Channel**. This creates a *Provided Event Channels* relationship between the Event Channel and the Application.

      You can see the relationship created by expanding the relationship folder within the Application.

When the Configuration is next activated, the event channel defined in this procedure will be set up on each server that the Application is configured on.

### Related Tasks

“Configure an Application onto a Freestanding Server” on page 186

## Configure a Client Style to use Session Services

Use this procedure to *configure* a client style to enable clients of that style to use session services. This enables client applications to start a session when they want to perform a set of activities as a single logical activity, a **unit-of-activity**.

Note that by default client styles are enabled to use session services.

**To configure a client style for session services, complete the following steps:**

1. Expand the Configuration that contains the Client Style model.
2. Expand the **Client Styles** folder, to display the Client Style model.
3. From the pop-up menu of the Client Style model, click **Edit**. This displays the Object Editor that you can use to ensure that the **session enabled** attribute is set to **yes**.
4. If you change the **session enabled** attribute to **yes**, click **OK** to apply the change and close the Object Editor window.

When the Configuration is next activated, the client style will be enabled to use the session services.

## Configure a Client Style to use a Specific Factory Finder

Use this procedure to configure a client style to enable clients of that style to use a specific factory finder for their transaction services, and other client services.

**Prerequisites:** The Factory Finder model that defines the factory finder to be used by the client style must already exist. If you want to create a new Factory Finder model, see “Configure a Factory Finder” on page 236.

**To configure a client style to use a specific factory finder, complete the following steps:**

1. Expand the Configuration that contains the Client Style model.
2. Expand the **Client Styles** folder, to display the Client Style model.
3. From the pop-up menu of the Client Style model, click **Edit**. This displays the Object Editor that you can use to change the **factory finder** attribute.
4. Change the **factory finder** attribute to specify the name of the factory finder. The value is the fully-qualified name path from the local root, which can be used in a resolve to get the factory finder desired. For example, `host/resources/factory-finders/xyzServer-server-scope`
5. To apply the change and close the Object Editor window, click **OK**.

When the Configuration is next activated, the client style will be enabled to use the specified factory finder.

### Related Tasks

“Configure a Factory Finder” on page 236
“Configure a Chain of Location Objects” on page 234
“Configure a Single Location” on page 231
“Edit Objects” on page 72

# Chapter 15. Operate your Enterprise

When you activate a Configuration of one of your Management Zones, the System Manager creates or updates the runtime configuration of that Management Zone in your enterprise. The runtime configuration is represented by the *Active Configuration* within the Management Zone and by Image objects within the Host Images for hosts used by the Management Zone.

The Active Configuration provides a view of the runtime configuration across all hosts used by the Management Zone. The Host Images provide a view of the runtime configuration on the one host. For more information, see “System Management Objects used to Operate your Enterprise” on page 45.

This topic describes some general tasks that you can use to operate your enterprise; to monitor and act on the runtime configuration of your enterprise. Any changes that these tasks make to the runtime configuration last only until the Active Configuration is activated again. Other tasks that affect the runtime configuration are also described in the other “How this Book is Organized” on page xiii.

If you want to make more long-term changes to the Active Configuration, you should change the system management Configuration within the Management Zone, then activate that Configuration again. For more information, see “Change the Active Configuration of Your Enterprise” on page 431.

If you want to investigate any problems that you may be experiencing with the runtime configuration of your enterprise, you should see the *Problem Determination Guide*.

## An Overview of Operating your Enterprise

Operation of an enterprise is generally aimed at ensuring that things keep running normally, and if needed taking actions to prevent problems occurring. Operation actions are generally short-term and targetted at specific objects; for example, to stop an application server or try it with a larger thread pool size.

Generally, you do not need to operate system management components, because they are started and stopped automatically.

The sorts of things that you might do when operating a Component Broker enterprise are as follows:

- Subscribe to and monitor events that affect Image objects.

  CBConnector System Management automatically records events that it has caused; for example, the starting of an asynchronous action, and the subsequent successful completion of that action. However, there are many more events that you can **Subscribe** to; for example, creation and deletion of objects, and any change in an object’s attributes.

  The System Manager user interface automatically displays messages for events that you have subscribed to. You can then click on the event message to display and act on the object affected.

  For more information, see “Monitor Events” on page 419.
- Display the runtime status of objects, in particular application servers.

When you select an Image object displayed on the System Manager user interface, the Status bar displays the runtime status of the object.

For more information, see “Display Server Health” on page 195.

- Use the File Browser to display information recorded in Component Broker activity, error, and trace logs for objects managed by CBConnector System Management.

  Component Broker automatically records runtime information in its logs for you to monitor your enterprise and to investigate any problems that may occur.

  For more information, see “Display Information in Component Broker Logs” on page 423.

- Use the Tivoli Event Manager windows to monitor Component Broker events.

  If you have turned on Component Broker event generation for Tivoli and installed and configured it correctly, Component Broker will send Tivoli events to the Tivoli event server whenever an entry is added to the Component Broker error log.

  For more information, see “Display Component Broker Events Using the Tivoli Event Manager” on page 424.

- WIN Use the Windows NT Event Viewer to monitor Component Broker messages in the Windows NT event log.

  Many applications running on Windows NT store messages in the Windows NT event log. All messages stored by the CBConnector product can be distinguished by the string **CBConnector** in the *Source* column in the Windows NT event log. Additional information indicating the type of message is stored in the Category column.

  To view the Windows NT event log you can use the Event Viewer, which is one of the Administration Tools (Common) for Windows NT.

- Operate Application Servers, Client Styles, and Applications.

  From time to time you may need to stop or start a specific application server, client style, or application running on a host managed by Component Broker. To do this, you can use the System Manager user interface to act on the appropriate Image object.

- Operate Communications Servers used for APPC connections to tier-3 systems.

  If any of your Component Broker application servers use APPC to connect to tier-3 systems across an SNA network, you are likely to need to act on the Communications Servers. To do this, you use the **SNA Node Operations** application provided with the Communications Server.

  For more information, see “Operating Communications Server” on page 429.

- Control the trace information recorded by Component Broker.

  You can turn on the Component Broker trace function to record details about the process flow for its components and object services. You can also enable or disable the ability to trace runtime problems back from servers to a specific client.

  Such tasks are normally only needed if a problem cannot be resolved using other runtime information and you have been advised to capture trace information (typically for use by IBM support personnel).

  For more information, see “Control Component Broker Trace” on page 426.

### Related Concepts

“System Management Objects used to Operate your Enterprise” on page 45
“The Image World” on page 40

“System Management Representation of Your Enterprise” on page 37
“Example: Operate an Application” on page 250
“Example: Operate a Server” on page 201
“Operating CBConnector System Management Components” on page 431

**Related Tasks**

“Monitor Events” on page 419
“Display Runtime Information about your Enterprise” on page 422
“Display Information in Component Broker Logs” on page 423
“Control Component Broker Trace” on page 426
“Change the Active Configuration of Your Enterprise” on page 431
“Chapter 6. Administer your Host Environment” on page 167
“Chapter 7. Administer Application Servers” on page 181
“Chapter 8. Administer Clients” on page 205
“Chapter 9. Administer Applications” on page 221
“Chapter 11. Administer Security in your Enterprise” on page 261
“Chapter 12. Administer Workload Management” on page 333
“Chapter 13. Administer Connections to Tier-3 Systems” on page 343
“Chapter 14. Administer Component Broker Services” on page 397

## Sources of Information

This topic outlines the sources of information that can help you to monitor the running of Component Broker environments. These sources of information can also be valuable in resolving problems and communicating with IBM support groups.

The sources of information are:

- Activity log (page 415)
- Error log (page 415)
- Trace log (page 416)
- Messages (page 416)
- User interface data (page 2 on page 417)

For more information about using these information source to resolve problems, see the *Problem Determination Guide*.

### Activity log

The *activity log* is used mainly for storing information about normal CBConnector program execution. This includes informational messages that show a history of activity of CBConnector services. This would be used by IBM Service personnel.

Another use of the activity log is for storing CORBA exceptions that are returned. This information can be used by IBM Service personnel to determine possible sequences of activity leading up to a problem. This information can also be used by CBConnector Application or Business Object developers during their program development.

There is one activity log for all CBConnector processes on each workstation that executes CBConnector code. It is stored as a file in the directory specified by the **activity log directory** attribute of the workstation’s Host Image. “Display Information in Component Broker Logs” on page 423.

### Error log

The *error log* is a facility that is used to store information when CBConnector detects an unexpected failure. This is an implementation of the "first failure data capture" philosophy. For example, error log entries are made for the following types of unrecoverable conditions:

- Assertion failure
- Unrecoverable error condition
- Vital resource failure, such as memory, semaphore, disk, and so on
- Operating system exceptions
- Programming defects in CBConnector code

There is one error log for all CBConnector processes on each workstation that executes CBConnector code. “Display Information in Component Broker Logs” on page 423.

### Trace log

The *trace logging service* is used to capture debug trace information in CBConnector code at runtime. This debug information contains execution path and data information. Separate controls are provided for each CBConnector component and varying levels of detail can be gathered. This service is for use exclusively by or on behalf of IBM Service personnel.

You can control whether all trace entries are stored in the same trace file or, each time the trace service is activated, it creates a separate trace file.

Copies of the error log and activity log entries may also be added to the trace logs when those entries are generated while trace is active. This provides an overall and correlated view of all known CBConnector activity while trace is active.

“Display Information in Component Broker Logs” on page 423.

### Messages

The following information is provided for messages in the activity log, error log, and trace log. For message descriptions, see the *Problem Determination Guide*.

**componentId**
The identifier of the CBConnector component that created the message entry.

**SOMProcessType**
This identifies which Component Broker process stored the entry. The values are:

1. Client
2. ORB daemon
3. SM agent
4. System Manager
5. SM EUI
6. Server

**processId**
The operating system process number that stored the entry in the log.

**threadId**
The operating system thread number that stored the entry in the log.

**functionName**
The name of the function that stored the message. In some cases the names of the class on the function name is provided. In other cases only the name of the source file is provided.

**probeId**
A unique identifier for the function that stored the message. In many cases this is the source file line number.

**sourceId**
Version information of the source file that contains the code that stored the message.

**rawData**
Additional data that can be used for problem determination.

**manufacturer**
The manufacturer of the executable code that stored the message. This is always "IBM"

**product**
The name of the product of the executable code that stored the message. This is always "CBConnector"

**version**
The version number of the product.

**serverName**
The name of the server that was executing the code that stored this message. This value is blank if the code that stored the message was not running in the server.

**clientHostName**
The name of the workstation that sent a request that resulted in this message being generated.

**clientName**
The name of the client workstation that was involved in the request in the server when this message was stored. This value is blank if the code that stored the message was not running in the server.

**category**
The category of this message, as indicated by the following values:

1. An error log message entry
2. An activity log message entry

**timeStamp**
The time that the message was stored.

**unitOfWork**
The complete path of workstations involved in processing a request from a client, in the form of *nnnn:clientName:server1name:server2name...* This information can be used to find all information on all workstations involved in a complicated request.

**severity**
The severity of the message. This can be **Informational**, **Warning**, or **Error**.

**primaryMessage**
The message that was stored.

**extendedMessage**
For some messages, additional information provided as part of the message. In many cases this is blank.

**User interface data**

When you use the System Manager user interface to start an asynchronous action, it displays an Action Console window to display messages about the action's progress. You can use Action Console window to monito the action and to check for successful completion.

As you use the System Manager user interface to open one system management object after another, it builds up a history list of those objects. You can use the History window to view the history list, to check the objects opened during the current session, display those objects, and to retrace the steps that you took to get to a particular object.

You can subscribe to changes in system management objects, to monitor changes to object attributes and relationships, and to monitor the creation and deletion of objects. You can use the Event Monitor window to see what events have occurred for objects that you have subscribed to.

## The Transaction Service Log

The Transaction Service creates a log for every server. This records information about transactions running in the server and is used for recovery.

**Important:**

The log here is not the same as a message log (such as the Component Broker activity log), audit trail, or journal service. It is only used for transaction state information and is not in a readable format.

The name of a server's log is assigned by the Transaction Service the first time the server is started, and appears in one of the messages written to the Component Broker activity log as the server starts up. The name is of the form somtr*nnn*; for example, somtr0000.

The Transaction Service log has two parts:

**The transaction log**
that records transactional state information for each transaction.

**The transaction-partner log**
that records information about partner systems involved in transactions; for example, partner systems such as DB2 servers or APPC partners.

The transaction log consists of three types of files (.t.ctl, .t.csh, .t.nnn) and the transaction partner log consists of three types of files (.p.ctl, .p.csh, .p.nnn), as follows:

**.ctl**
A control file that requires 13 KB (for example, somtr000.t.ctl and somtr000.p.ctl).

| | |
|---|---|
| **.csh** | A cushion file (for example, somtr000.t.csh and somtr000.p.csh). |
| **.nnn** | Any number of extent files (for example, somtr000.t.001, somtr000.t.002, and so on). The number of extent files used is dependent on the number and age of the running transactions in the server. |

For the transaction log, the size of the cushion file (.t.csh) and each extent file (.t.nnn) is defined, as a multiple of 64KB, by the **log file size** attribute for the server. For the transaction partner log, the size of the cushion file (.p.csh) and each extent file (.p.nnn) is fixed as 64KB.

Therefore the minimum size for both the transaction log files and the transaction partner log files is 141KB. However, depending on the number of transactions running in the server and their duration, you should expect a server to have 2 to 3 extent files at any one time. This would require a minimum of 269KB to 333KB.

The location of the log files is defined on the **log directory** attribute for the server. The directory specified must exist and be on a local file system. It is also advisable to use a file system where any other data on it is reasonably static. This will ensure there is always sufficient disk space for the log.

You can edit the **log directory** and **log file size** attributes on the Transaction Service notebook page for a server. The value of the **transaction service log file** attribute is displayed only on the Transaction Service page of the Server Image.

When a transaction completes running within a server, its entry is removed from the log. There is no need to perform housekeeping functions on the log files. If a server fails because of insufficient space in the file system, make more space without deleting any of the log files. The only time it is safe to delete a log file is if the file is for a server that has been deleted.

A server stores the name of its log in the Component Broker CDS. If you recreate your CDS, the server will no longer have the name of the log file and will be allocated a new log name. Therefore, before recreating your CDS, start up each server and ensure the Transaction Service starts successfully. Then, without allowing any new transactions to be started, shut the servers down. There will then be no partially complete transactions in your servers and you can safely delete the log files and recreate the CDS.

## Monitor Events

CBConnector System Management automatically records events that it has caused; for example, the starting of an asynchronous action, and the subsequent successful completion of that action. It records such events in the activity log for the SM agent on each managed server host. You can display such event messages by browsing the activity log.

However, there are many more events that you can **Subscribe** to; for example, any change in an object’s attributes. You can monitor events by displaying the events log, changing your event subscriptions, and using the event log to open objects for which an event has been recorded.

- Create or Change Event Subscriptions (page 420)
- Delete Event Subscriptions (page 421)

- Display The Events Log (page 420)
- React To Events (page 420)
- Remove An Entry From The Log File (page 420)

CBConnector System Management records events that you have subscribed to in the *event log* for the Information Controller window, and displays such events through the related Event Monitor window.

The events log file contains textual information about events that have occurred and actions that the user has taken during the current session. Each log message includes the time, date, type of action or event, the objects involved, and any system responses. For example:

```
06/05/97 10:37:36 :: subscription : attribute "operating mode" changed
from development to production :
object larner<Host Images><larner><Server Images><Sample Server 1>
```

lines above to make it easier to read.

### Display The Events Log

The events log file is displayed by the main Event Monitor window. To display that window, select the Event Monitor icon of the Tool bar.

### React to Events

When an event occurs, and CBConnector System Management displays the Event Monitor window you can use the **Go To** action on an entry in that window to go to the related object. You can then use the standard generic actions, such as **Edit** to display details about the object and to act on the object; for example to **Stop** a server.

### Remove an Entry from the Log File

Use this procedure to remove an entry about an event from the log displayed by the Event Monitor window. In the Event Monitor window.

1. Click on the event entry to be removed.
2. Click on the **Remove** button.

### Create or Change Event Subscriptions

Use this procedure to subscribe to an event for an object, or to change an event subscription.

1. Display the object
2. Click the right mouse button on the object to display its pop-up menu
3. Click on the **Subscriptions** menu choice to display the Events Subscriptions window for that object.

   In the the Events Subscriptions window:
4. Click on the **Create** button to create a new event subscription or the **Edit** action to change an event subscription. This displays the Subscriptions Editor window, which you can use to act on your event subscriptions.

   In the the Subscriptions Editor window:
5. Select the details of the event subscription using the following fields:

**Subscription Type field**

Use the Subscription Type radio button field to select the type of subscription to be created, as one of:

**Creation**

Issue an event when an object of this class gets created. (This can be selected for folders only.)

**Deletion**

Issue an event when the object is deleted.

**Relation**

Subscribe to an event on one of the object's relations.

**Attribute**

Subscribe to an event on one of the object's attributes.

**Relations / Attributes list**

Use the Relations / Attributes list to select a relation or attribute for which an event subscription is to be created. If the **Subscription Type** radio button is set to **Relation**, this panel lists the relations of the object. If the **Subscription Type** radio button is set to **Attribute**, this panel lists the attributes of the object. Otherwise, this field is blank.

**Attribute field**

Use the Attribute field to specify the type of event that you want to subscribe to for an attribute. Specify the type of event by selecting the following buttons:

**Change only**

Issue an event for any changes to the attribute value.

**Change and From**

Issue an event if the attribute value changes *from* the value specified in the **From** entry field.

**Change and To**

Issue an event if the attribute value changes *to* the value specified in the **To** entry field.

**Change, From, and To**

Issue an event if the attribute value changes *from* the value specified in the **From** entry field *to* the value specified in the **To** entry field.

**Event Monitor Action field**

Use this field to specify what the Event Monitor should do if the event being subscribed to does occur. Specify the action as one or both of:

**Highlight Event Entry**

Highlight the event entry in the message log.

**Popup Event Monitor**

Display the Event Monitor window.

6. When you have completed your changes, select one of the following buttons at the bottom of the window:

**OK** Save any changes and close the window.

**Cancel**

Cancel any changes and close the window.

### Delete Event Subscriptions

Use this procedure to delete a subscription to an event for an object.

1. Display the object
2. Click the right mouse button on the object to display its pop-up menu
3. Click on the **Subscriptions** menu choice to display the Events Subscriptions window for that object.

   In the Events Subscriptions window:
4. Click on the event subscription in the information panel.
5. Click on the **Delete** button.

**Related Concepts**

“Event Monitoring” on page 8
“Event Subscriptions” on page 443

**Related Tasks**

Display the Events Log (page 420)
“Example: Operate a Server” on page 201

## Display Runtime Information about your Enterprise

When you select an Image object on the System Manager user interface, the status line displays a summary of runtime information about that object. For example, if you select a Server Image the Status bar would display information like the following:

```
My Server 1:  runOnRequest running 3 3 good
```

This indicates that the server called **My Server 1** is running with good health. For more information, see “Display Server Health” on page 195.

Component Broker provides a File Browser that you can use to display information recorded in activity, error, and trace logs for objects managed by CBConnector System Management.

If you have enabled Tivoli to monitor Component Broker events and installed and configured it correctly, you can use the Tivoli Event Manager windows to display Component Broker events that were sent to Tivoli.

If your Component Broker applications running on Windows NT store messages in the Windows NT event log, you can use the Windows NT Event Viewer.

For more information about using these tools, see the related topics.

**Related Concepts**

“Problem Determination” on page 471
“Integrated System Management with Tivoli” on page 8
Messages (page 416)

**Related Tasks**

“Display Server Health” on page 195
“Display Information in Component Broker Logs” on page 423
“Display Component Broker Events Using the Tivoli Event Manager” on page 424
“Display Component Broker Messages in the Windows NT Event Log” on page 425

"Install and Configure the Component Broker plus module for Tivoli" on page 47
"configure and enable Tivoli event monitoring" on page 50

## Display Information in Component Broker Logs

You can use the *"File Browser" on page 449* to display information recorded in activity, error, and trace logs for objects managed by CBConnector System Management. This helps administrators to identify any problems recorded in the log files. It provides a range of functions to help view the file contents; for example, it can be used to move around the file, change the format of the display, filter the display, and search for entries.

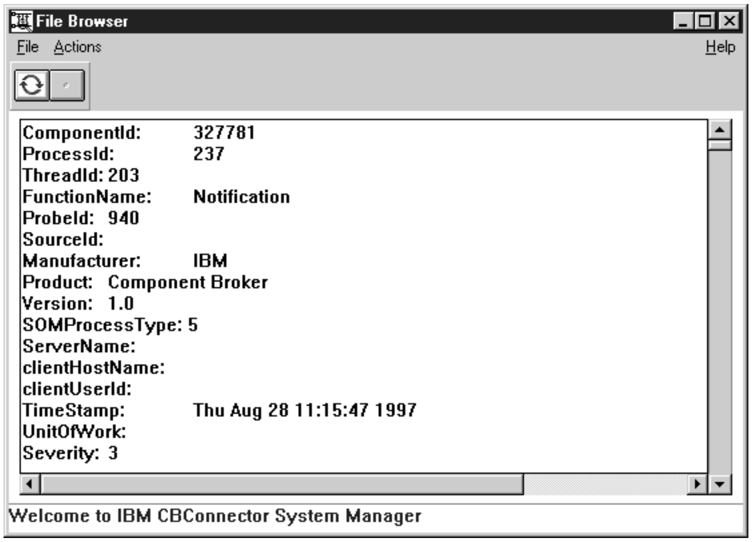
To display a File Browser window, click **Browse** on the pop-up menu of a log file object, which you can display by expanding the object that they are contained in. This displays a File Browser window, as shown in Figure 71.

Log information can be displayed in a series of columns (like the Windows NT Event Viewer) or as plain text. To toggle between column layout and plain text, click on **Actions - Column layout**.

In column layout, the File Browser displays basic information about all the entries in the log. To display more details about an entry in the log, double-click on the entry. (Alternatively, select the entry then click **Actions - View Detail**.) This displays a dialog window containing the detailed information.

To move around the information displayed, and to search for specific information, use the window scroll bars and the **Actions** menu-bar choice. For example, to search for a specific string, click **Actions - Search**.

To close a File Browser window, click the close icon in the top right corner of the window.



```
ComponentId:        327781
ProcessId:          237
ThreadId: 203
FunctionName:       Notification
ProbeId:  940
SourceId:
Manufacturer:       IBM
Product:  Component Broker
Version:  1.0
SOMProcessType: 5
ServerName:
clientHostName:
clientUserId:
TimeStamp:          Thu Aug 28 11:15:47 1997
UnitOfWork:
Severity: 3
```

Welcome to IBM CBConnector System Manager

*Figure 71. The File Browser window*

### Related Concepts

“File Browser” on page 449
Messages (page 416)
“Problem Determination” on page 471

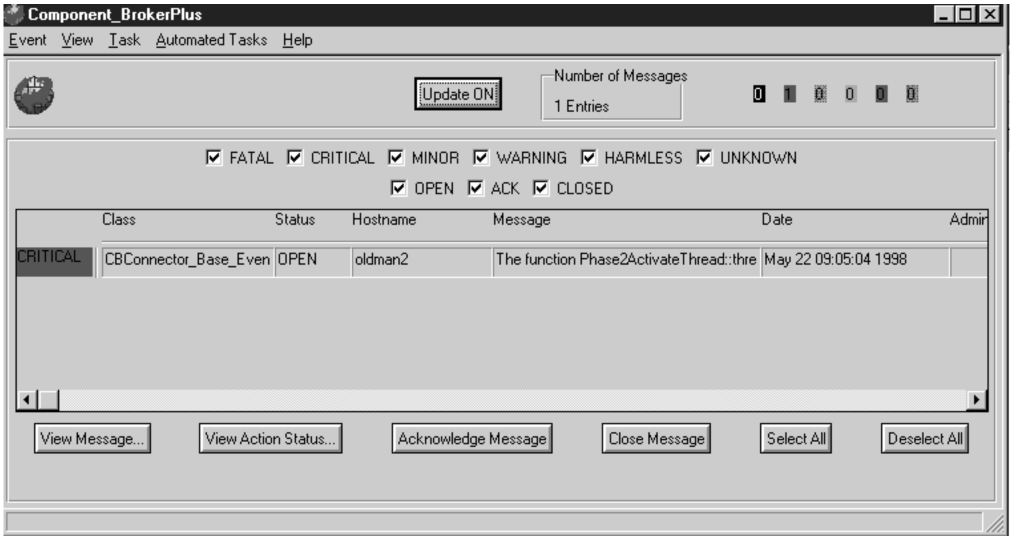
### Related Tasks

Display the Activity Log
“Display Component Broker Events Using the Tivoli Event Manager”
“Display Component Broker Messages in the Windows NT Event Log” on page 425
“Control Component Broker Trace” on page 426

## Display Component Broker Events Using the Tivoli Event Manager

If you have turned on Component Broker event generation for Tivoli and installed and configured it correctly, Component Broker will send Tivoli events to the Tivoli event server whenever an entry is added to the Component Broker error log.

You can use the Tivoli Event Manager windows to display such Component Broker events, as in the following example:



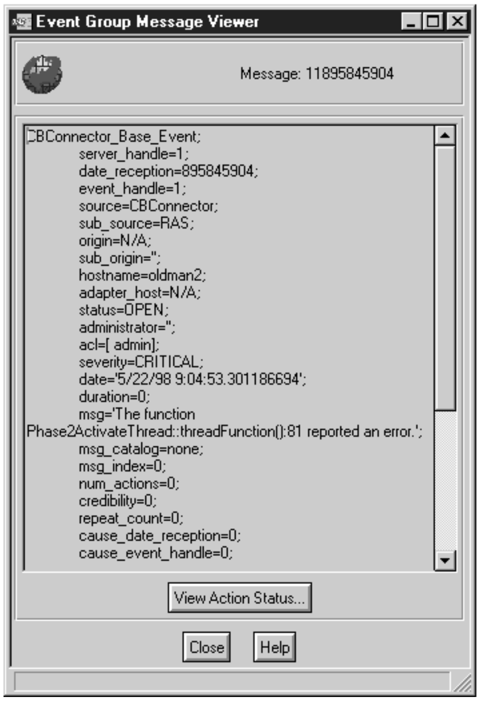
Number of Messages: 1 Entries

0 1 0 0 0 0

☑ FATAL ☑ CRITICAL ☑ MINOR ☑ WARNING ☑ HARMLESS ☑ UNKNOWN
☑ OPEN ☑ ACK ☑ CLOSED

| | Class | Status | Hostname | Message | Date | Admin |
|---|---|---|---|---|---|---|
| CRITICAL | CBConnector_Base_Even | OPEN | oldman2 | The function Phase2ActivateThread::thre | May 22 09:05:04 1998 | |

*Figure 72. The Tivoli Event Manager main window.*
*This shows the Tivoli Event Manager main window with a Component Broker event.*



```
CBConnector_Base_Event;
        server_handle=1;
        date_reception=895845904;
        event_handle=1;
        source=CBConnector;
        sub_source=RAS;
        origin=N/A;
        sub_origin='';
        hostname=oldman2;
        adapter_host=N/A;
        status=OPEN;
        administrator='';
        acl=[ admin];
        severity=CRITICAL;
        date='5/22/98 9:04:53.301186694';
        duration=0;
        msg='The function
Phase2ActivateThread::threadFunction():81 reported an error.';
        msg_catalog=none;
        msg_index=0;
        num_actions=0;
        credibility=0;
        repeat_count=0;
        cause_date_reception=0;
        cause_event_handle=0;
```

*Figure 73. A Tivoli Event Detail window.*
*This shows the Tivoli Event Detail window for a Component Broker event.*

To display the Tivoli Event Manager windows, see the information provided with your Tivoli product.

**Related Concepts**

“Problem Determination” on page 471
Using Tools For Problem Determination
“Integrated System Management with Tivoli” on page 8
Messages (page 416)

**Related Tasks**

“Display Information in Component Broker Logs” on page 423
“Display Component Broker Messages in the Windows NT Event Log”
“Control Component Broker Trace” on page 426
“Install and Configure the Component Broker plus module for Tivoli” on page 47
“configure and enable Tivoli event monitoring” on page 50

## Display Component Broker Messages in the Windows NT Event Log

WNT Many applications running on Windows NT store messages in the Windows NT event log. All messages stored by the CBConnector product can be distinguished by the string **CBConnector** in the *Source* column in the Windows NT event log. Additional information indicating the type of message is stored in the Category column.

To view the Windows NT event log you can use the Event Viewer, which is one of the Administration Tools (Common) for Windows NT.

**Related Concepts**

"Problem Determination" on page 471
Messages (page 416)

**Related Tasks**

"Display Information in Component Broker Logs" on page 423
"Display Component Broker Events Using the Tivoli Event Manager" on page 424
"Control Component Broker Trace"

## Control Component Broker Trace

CBConnector System Management automatically records trace information for Component Broker object services. You can control the amount of information recorded by setting the level of trace on attributes of various system management objects.

To set the trace level for a system management object, you generally:

1. Display the object in an Information Controller window
2. Use the **Edit** action to display the Object Editor
3. Click the **Component Trace** or **Object Services Trace** notebook tab
4. Change the required trace level settings
5. Click the **OK** button to apply the changes and close the Object Editor

The following trace levels can be set:

| Component Trace page | Object Services Trace page |
|---|---|
| caching component<br>heap component<br>iom component<br>application adaptor business object<br>application adaptor framework<br>XA framework<br>DB2 application adaptor<br>Oracle application adapter<br>root application adapter<br>main<br>memory component<br>NLS component<br>orb communications<br>orb IR<br>orb mutex<br>orb request<br>PAA (Procedural Application Adapter)<br>PAA communications<br>semaphores component<br>server groups component<br>threading component<br>utilities component | concurrency component<br>events component<br>externalization service<br>identity service<br>life cycle service<br>naming service<br>query service<br>security service<br>session services<br>transaction service<br>workspace cache |

Each of these attribute can be set to one of the following values, where each succeeding value increases the amount of information that is captured.

**Note:** Not all object services support all trace levels, or provide the same level of information. You should only set these trace levels as advised by IBM support personnel.

**none** Trace data is not recorded for this component.

**basic** The smallest amount of trace information, critical path trace data, is recorded. This data is primarily used for the highest level data and performance measurements.

**intermediate**
Record trace messages and any throw instructions that are processed, in addition to the information recorded for the **basic** trace level.

**advanced**
Record all trace information, including process flow and detailed data, in addition to the information recorded for the **intermediate** trace level. Further, messages sent to the activity and error logs are also recorded in the trace log. This data is primarily for extended problem determination. It controls the recording of extra raw data, component extended messages, and indications that an exception subclass was thrown.

**Default:** none

**Related Concepts**

Trace log (page 416)
"Sources of Information" on page 415
"Problem Determination" on page 471

## Disable or Enable RAS Request Interceptors

RAS request interceptors can enable you to trace runtime problems back from servers to a specific client.

By default, RAS request interceptors are enabled in all clients and servers by the **enable RAS request interceptors** attribute of Client Styles, Server Groups, and Servers (freestanding). With RAS request interceptors enabled, log entries (and trace entries if you have turned trace on) have additional 'UnitOfWork' information that can help you debug runtime problems. But that information comes with a performance penalty. When your application environment is fully debugged and deployed, you should consider turn off RAS interceptor filters to imrpove performance.

You can turn off RAS request Interceptors to improve the performance of your Component Broker application environment, but only if you are confident that the environment has been stable for some time and that you need some extra performace. You can turn RAS request interceptors off and on in any combination of servers and clients. For maximum performance improvement, you should turn off RAS request interceptors in your Client Styles, Servers, and Server Groups.

By turning off the filters on a client, the server-side RAS trace and activity log entries will not contain the data needed to map an event back to that specific client. If you later decide that you need that information, turn on the filters in the Client Style then restart the client.

By turning off the filters on the server side, no RAS trace and activity log entries will be mappable back to any specific client. If you later decide that you need that information, turn on the filters in the Server (or Server Group) then restart the server. **Note:** You should turn these filters on and off in the Client Styles, Servers, and Server Groups of your system management Configuration. After you next activate that Configuration, check that your clients and servers are restarted to implement the change.

To disable or enable (set by default) RAS request interceptors, complete the following steps:

1. Expand the Management Zone that defines your application environment
2. Expand the Configurations folder
3. Expand the Configuration within which the server group is defined
4. To change the RAS request interceptors for a server group, complete the following steps:
   a. Expand the Server Groups folder
   b. From the pop-up menu of the Server Group model, click **Edit**. This displays the Object Editor window for the Server Group model.
   c. In the Object Editor Notebook, click the **Log Controls** tab.
   d. Change the setting of the **RAS request interceptors** attribute as appropriate
   e. To save the changes and close the Object Editor notebook, click the **OK** button.
5. To change the RAS request interceptors for a freestanding server, complete the following steps:
   a. Expand the Servers (freestanding) folder
   b. From the pop-up menu of the Server (freestanding) model, click **Edit**. This displays the Object Editor window for the Server (freestanding) model.
   c. In the Object Editor Notebook, click the **Log Controls** tab.
   d. Change the setting of the **RAS request interceptors** attribute as appropriate
   e. To save the changes and close the Object Editor notebook, click the **OK** button.
6. To change the RAS request interceptors for a client, complete the following steps:
   a. Expand the Client Styles folder
   b. From the pop-up menu of the Client Style model, click **Edit**. This displays the Object Editor window for the Client Style model.
   c. In the Object Editor Notebook, click the **Log Controls** tab.
   d. Change the setting of the **RAS request interceptors** attribute as appropriate

      **yes** Log entries (and trace entries if you have turned trace on) have additional ’UnitOfWork’ information that can help you debug runtime problems. But that information comes with a performance penalty. When your application environment is fully debugged and deployed, you should consider disable RAS interceptor filters to imrpove performance.

      **no** Log entries (and trace entries if you have turned trace on) do not have ’UnitOfWork’ information.

      By disabling the filters on a client, the server-side RAS trace and activity log entries will not contain the data needed to map an event back to that specific client. If you later decide that you need that information, enable the filters again in the Client Style then restart the client.

By disabling off the filters on the server side, no RAS trace and activity log entries will be mappable back to any specific client. If you later decide that you need that information, enable the filters again in the Server (or Server Group) then restart the server.

**Default:** Yes

**Note:** You should disable or enable these filters in the Client Styles, Servers, and Server Groups of your system management Configuration. After you next activate that Configuration, check that your clients and servers are restarted to implement the change.

e. To save the changes and close the Object Editor notebook, click the **OK** button.

7. On the pop-up menu of the Configuration, click **Activate** to update your application environment.

   Any changes made will be implemented the next time that clients and servers are restarted.

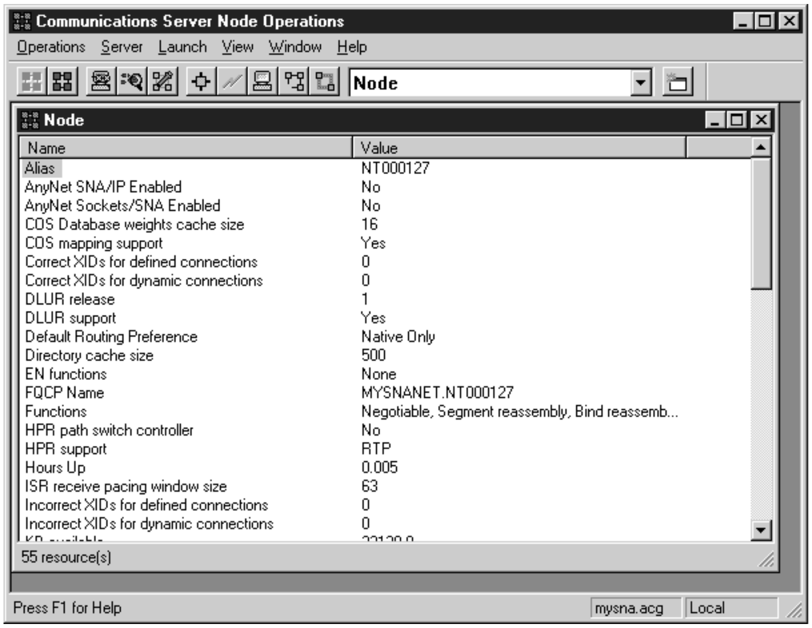
## Operating Communications Server

If any of your Component Broker application servers use APPC to connect to tier-3 systems across an SNA network, you are likely to need to act on the Communications Servers.

This topic provide an overview of operating Communications Server. For more details, and task descriptions, see *Using IBM Communications Server for Windows NT with CICS*, SC33-1900.

Communications Server resources are managed using the **SNA Node Operations** application, provided with the Communications Server. The main window is shown in the figure SNA Node Operations application (page 429).

**SNA Node Operations application**



Communications Server Node Operations
Operations Server Launch View Window Help
Node
Node
55 resource(s)
Press F1 for Help
mysna.acg
Local

| Name | Value |
|---|---|
| Alias | NT000127 |
| AnyNet SNA/IP Enabled | No |
| AnyNet Sockets/SNA Enabled | No |
| COS Database weights cache size | 16 |
| COS mapping support | Yes |
| Correct XIDs for defined connections | 0 |
| Correct XIDs for dynamic connections | 0 |
| DLUR release | 1 |
| DLUR support | Yes |
| Default Routing Preference | Native Only |
| Directory cache size | 500 |
| EN functions | None |
| FQCP Name | MYSNANET.NT000127 |
| Functions | Negotiable, Segment reassembly, Bind reassemb... |
| HPR path switch controller | No |
| HPR support | RTP |
| Hours Up | 0.005 |
| ISR receive pacing window size | 63 |
| Incorrect XIDs for defined connections | 0 |
| Incorrect XIDs for dynamic connections | 0 |

Underneath the menu bar are a number of buttons that duplicate the most popular options provided by the menu. To the right of the buttons is a drop-down menu that you can use to see a list of resources to act on.

The option you select determines the type of resources that are displayed in the main window. The most useful options when running Communications Server with Component Broker are:

**Connections**
displays the status of the connections in your SNA network.

**Local LU 6.2**
displays information about the local LUs (Component Broker application servers).

**LU 6.2 Sessions**
displays information about each session that is active between your application servers and remote tier-3 systems.

**Modes**
displays information about the modenames that are in use.

**Node**
displays information about the local host, on which your application servers (and Communications Server) are running.

**Partner LU 6.2**
displays information about the remote tier-3 systems that are in use.

Before you can view these resources you must start the node and load your configuration file into the **SNA Node Operations** application.

When the node is started, Communications Server will try to start a number of NT services and devices. It may also try to contact remote systems if the connection to them is defined as **Auto-Activate**. If there are any errors on the configuration file, or if one of these remote systems is unavailable, Communications Server will write error messages to its log file. This can be viewed using the **Log Viewer** application which can be started from the **Launch** menu of the **SNA Node Operations** application.

When the node successfully starts, details of your local machine are displayed in the main window of the **SNA Node Operations** application.

You can then use the central menu to view the other resources you have defined for SNA.

**Related Concepts**

“IBM Communications Server” on page 34
“Introduction to SNA” on page 471
“Connections to Tier-3 Systems” on page 29

**Related Tasks**

“Collect information for your SNA configuration” on page 358.
“Configure Communications Server” on page 367
“Configure an APPC Connection to a Tier-3 System for use by Applications” on page 354

## Operating CBConnector System Management Components

The System Manager or SM Agent on a host has its own process that runs as a Windows NT service or AIX resource. The process is started automatically when the host is started, but you can stop and start the process at other times. Normally, there should be no need to stop and start the process manually.

WIN The Windows NT service is called *CBConnector*.

**Related Concepts**

“Components used for System Management” on page 9

**Related Tasks**

“Start or Stop the System Manager User Interface” on page 55
“Verify that System Management Components are Running” on page 169

## Change the Active Configuration of Your Enterprise

The active configuration of your enterprise is represented and operated by the Images that exist on managed hosts. Those Images are related to model objects contained in the *Active Configuration* of each Management Zone.

To change the active configuration of your enterprise, you change the models in a Configuration of a Management Zone, then activate that Configuration again.

For example, to add a new application to a running server, you complete the following actions:

1. Define the new Application in the appropriate Configuration of your application Management Zone
2. Edit the attributes and relationships of the Application
3. Configure the Application onto a Server Group or Server (freestanding)
4. Activate the Configuration

Likewise, to delete an application server from a server group, you complete the following actions:

1. Delete the Server (member of group) from the Configuration
2. Activate the Configuration

In this second example, the System Manager would stop the application server on the host on which it was running. However, to permanently delete all trace of the server from the host, you must then **remove** the Server Image, as described in "Remove an Application Server from a Host" on page 196. Also, the Applications configured onto the Server Group would still be used by other Servers (member of group) within that server group.

### Related Concepts

"The Model World" on page 39
"The Image World" on page 40

### Related Tasks

"Remove an Application Server from a Host" on page 196
"Activate a Configuration" on page 256
"Delete Objects from an Active Configuration" on page 259

# Appendix A. Features of the User Interface

This topic describes the features of the CBConnector System Manager user interface.

Effective system management depends on being able to act on any part of your enterprise quickly and easily from a single point of control. This in turn depends on a standard interface that presents a clear picture of the enterprise and provides easy to use functions to act on the enterprise. To go beyond simple operation of a enterprise, the user interface has to be able to represent the enterprise as you would like to administer it.

The CBConnector System Manager user interface provides the following features to enable effective systems management:

**"Standard Windows" on page 434**
Windows that have standard control features familiar to users of other graphical user interfaces.

**"Using The Mouse And Keyboard" on page 436**
Mouse and keyboard actions, such as point and click, enable users to act on the user interface graphically as with other graphical user interfaces.

**"Graphical Presentation Of Objects" on page 437**
A network of standard objects displayed as icons that you can navigate around and affect by actions selected from menus and other graphical tools.

**"The Object Editor Window" on page 439**
A graphical editor for you to display and change the attributes of objects on any host managed by the System Manager.

**"Session History" on page 441**
A record of every object that you opened during the current user interface session only and from which you can display selected objects directly.

**"Hotlist" on page 442**
A list of objects that are especially important to you that is preserved across all user interface sessions and from which you can display selected objects directly.

**"Event Subscriptions" on page 443**
Functions to help you to subscribe to possible events, display system events and subscribed events, and to react to events that occur.

**"User-Level Settings and Object-Level Filters" on page 438**
Functions to help you filter the objects displayed by the user interface, to limit the display to the objects that you are especially interested in.

**"Wizards" on page 447**
The System Manager user interface provides wizards that you can use to simplify the related tasks. For example, you can use a wizard to create a server group, including defining the servers that are members of the group, and configuring the group as a controlled server group for workload management.

**"Help Information" on page 449**
Information to help you use CBConnector System Management, about the user interface, the objects it displays, and the actions that you can take.

**“File Browser” on page 449**
A tool that you can use to display the activity, error, and trace logs produced by CBConnector, to help you monitor your enterprise and resolve any problems that may occur. This can also be used to display and act on the file systems on hosts managed by CBConnector System Management

**Related Concepts**

“Standard Windows”
“Using The Mouse And Keyboard” on page 436
“Graphical Presentation Of Objects” on page 437
“The Object Editor Window” on page 439
“Session History” on page 441
“Hotlist” on page 442
“Event Subscriptions” on page 443
“User-Level Settings and Object-Level Filters” on page 438
“Help Information” on page 449
“File Browser” on page 449

**Related Tasks**

“Start the System Manager User Interface on Windows NT” on page 56
“Exit the User Interface” on page 60
“Chapter 2. Use the System Manager User Interface” on page 55

## Standard Windows

The CBConnector System Manager user interface is based on a set of windows that have standard elements common to many graphical user interfaces:

- Title bar
- Title bar icon
- Maximize button
- Minimize button
- Close icon
- Information panel
- Scroll bars
- Menu bar
- Tool bar
- Push buttons
- Pop-up menus
- Status bar

Figure 74 on page 435 shows the Information Controller window, the main window of the System Manager user interface, and its standard elements.

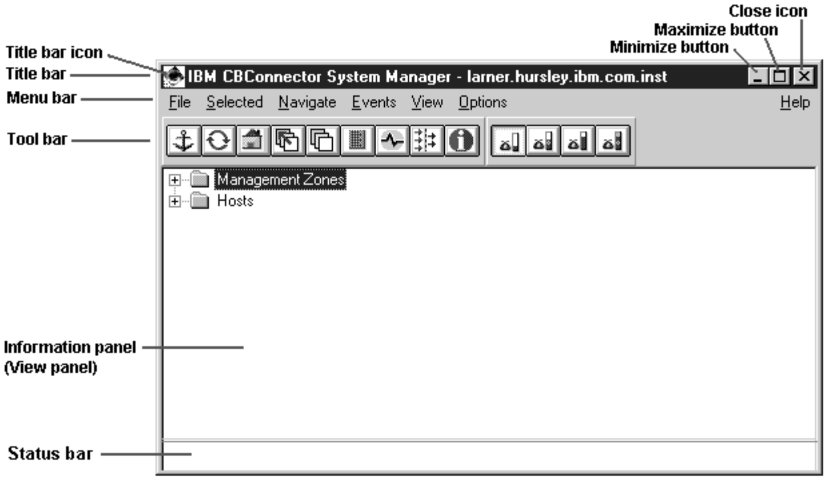


*Figure 74. The Information Controller window showing standard window features.*

The **Title bar** indicates the subject of information displayed in the window. You can use the Title bar, the **title bar icon**, the **maximize button**, and the **minimize button** to change the size and position of the window. You can use the **close icon** to close the window and exit the SM user interface.

You can use the **View panel** to view and act on information about the current object. Where the amount of information cannot be all displayed in the panel, it has scroll bars that you can use to move around the information.

The **status bar** displays "floating" information about areas of the window and objects displayed within it, and can display other information; for example, the state of a server. As you move the mouse pointer around the window, the status bar is dynamically updated with information about what is pointed at. You can display more detailed information about an object or an area of a window by using the **Help** function; for example, from the **Help** menu bar choice. (For more information about the help function, see “Help Information” on page 449.)

You can use the **menu bar**, **tool bar**, **push buttons**, and **pop-up menus** to act on the window, the current object, and the information displayed. To display a pop-up menu move the mouse pointer onto the required object or window area then press the right mouse button.

### Related Concepts

“Appendix A. Features of the User Interface” on page 433
“Session History” on page 441
“Hotlist” on page 442
“Event Subscriptions” on page 443
“User-Level Settings and Object-Level Filters” on page 438
“Help Information” on page 449
“File Browser” on page 449

### Related Tasks

“Chapter 2. Use the System Manager User Interface” on page 55

## Using The Mouse And Keyboard

You use the mouse and keyboard to act on the user interface, to affect the windows of the interface and the objects displayed.

The mouse and keyboard actions that you can use on a window are detailed in the **Keys Help** choice of the online help, selected from the **Help** menu bar choice.

Some of the more common mouse and keyboard actions are:

- To expand an object’s branch in a tree structure, click the left mouse button on the "+" to the left of the object. (This changes the "+" to a "-".) Click the left mouse button on a "-" to contract the object’s branch.
- To select an object or window area for a subsequent action or to select the action of a push-button or icon in the Tool bar, **click** once the left mouse button on the object or area.
- To use the default action of a window area, **double-click** the left mouse button on the object or area. For example, in Tree view, if you double-click on an object, the tree is expanded from that object.
- To display the pop-up menu for an object or window area, click the right mouse button on the object or window area. Move the mouse to point to the required action, then click the left mouse button to select the action. The action is used on that one object or window area.

You can use the keyboard to type text into fields of the user interface (for example, in attribute fields in the Object Editor window) and to help select and deselect objects and window areas.

- Press the Tab key to move the **focus** to the required area of the window. The focus marks the window area that can be acted on by the keyboard.
- Press the arrow keys ↑←↓→ to move the focus to the required object in the window area.
- Press the Shift+F8 keys to start extended selection of multiple objects. Use the arrow keys to move the focus to an object, then press the Spacebar to select the object. To add other objects to the selected group, move the focus to another object and then press the Spacebar.

### Related Concepts

“Appendix A. Features of the User Interface” on page 433
“Chapter 2. Use the System Manager User Interface” on page 55

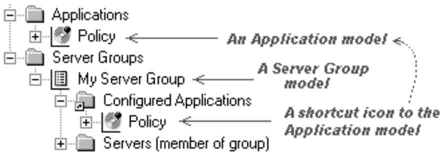
### Related Tasks

“Display Objects” on page 64
“Select and Deselect Objects” on page 65
“Act on Objects” on page 66

## Graphical Presentation Of Objects

CBConnector System Management represents your enterprise and its system management components as a network of objects connected by relationships. The network is one instance of the CBConnector System Management “System Management Representation of Your Enterprise” on page 37. You can move from one object to another, and the user interface updates itself automatically with details about the **current object**.

The Information Controller window displays the objects related to the current object. Objects are displayed as icons labeled by their names. For example: model objects for an application called **Policy** configured onto the server group called **My Server Group** are represented by the following icons:



The icons indicates the **class** of object, which in this case are a *Server Group* model and an *Application* model. Note that the Application model is in a *relationship* folder, indicated by an arrow mark on the bottom left of the folder.

Relationships are represented by shortcut icons. For example, in the figure above the **Policy** icon is a shortcut icon to the Application model called **Policy**, which is contained in the *Applications* folder. The **Policy** icon represents one instance of the relationship called *Configured Applications*, which defines that the applications are to be run on servers that are members of the server group. You can use relationships to easily navigate between objects and, from one object, to act on its related objects.

Some objects contain other objects for things that are logically "on" or "in" them. For example, a *Server Group model* contains *Server (member of group)* model objects for all the servers that are members of the server group, as shown in the figure above.

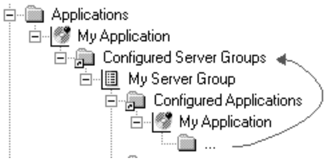
Objects that are folders (used *only* to contain other objects) generally can contain objects of one class only.

You can act on an object easily by moving the mouse to point at the object then clicking the right mouse button to display a pop-up menu of appropriate actions. For example, you can select **Drag**, display another object, then in the second object’s menu, select a context-sensitive **Drop** action involving the two objects. In most cases, dragging and dropping objects creates *relationships* between objects.

The object network can be displayed as a tree structure, with objects at one level of the tree containing other objects lower down the same branch. Objects also have relationships that go across the tree structure to other objects. The top level of the tree structure is called the **Home** view. When you navigate around the object network, you effectively move down, up, and across the tree structure.

You can either **expand** or **open** objects in the tree structure. If you expand an object, the next level of objects that it contains are added to the tree, and the current object does not change. If you open an object, you effectively move down the tree. The object that you have opened is called the **current object**. If you open a relationship you move across the tree structure, and can move up or down, to another object, which becomes the current object.

If you expand a branch of the tree view several times, you are likely to come to an object that you have already displayed in that branch. This has the potential to create an unwelcome loop back to that earlier object. To prevent this, the next icon in the branch is labelled by an ellipsis (...) and prevents further expansion of the branch. For example, the ellipsis in the following figure refers back to the objects in the same branch as indicated by the red arrow.



```
Applications
  My Application
    Configured Server Groups
      My Server Group
        Configured Applications
          My Application
            ...
```

The significant things about navigating around the object network are that:

- Only if you open an object is that object added to the session history list; if you simply expand an object in the tree structure, the history list does not change.
- If the View panel *is* displaying a tree structure, opening an object displays a new tree structure that starts with the contents of that object.
- If the View panel *is not* displaying a tree structure, opening an object displays only the contents of that object.
- You can use actions to go **back** along your path around the network, **up** the tree structure, or directly back to the Home view.

### Related Concepts

“Appendix A. Features of the User Interface” on page 433
“Standard Windows” on page 434
“Chapter 2. Use the System Manager User Interface” on page 55

### Related Tasks

“Control Which Objects are Displayed” on page 62

## User-Level Settings and Object-Level Filters

You can use *user levels* and *filters* to focus your view on subsets of the objects displayed in the View panel of an Information Controller window. Each Information Controller window has its own user-level setting and object-level filter.

**User-level setting**

The user-level setting controls what can be displayed for the user of an SM user interface, and therefore controls what that user can act on. It has the following four levels for different types of users, listed from the most-restrictive to least-restrictive levels:

**Basic**

This displays enough model objects for most applications to be configured and run, and enough image objects for servers and applications to be started and stopped, and statistics gathered.

**Advanced**

This displays everything in the Basic view and extra model objects to allow tuning, and more image objects to allow detailed statistics to be gathered.

**Expert**

This displays everything in the Advanced view and all image objects; enough for problem determination.

New users are recommended to use the most restrictive user-level setting that meets their needs.

Many objects, relationships, and attributes are protected against change by anyone without the appropriate user-level setting. If you need to change an object that is protected, you must first “Control Which Objects are Displayed” on page 62 to the required level. If you do this, be careful how you act on objects and, when finished, reset the user-level setting back to a suitable more-restrictive value.

To change the user-level setting, click one of the options under **View - User Level**.

**Object-level filter**

The object-level filter controls what can be displayed *within the bounds of the current user-level setting*. Therefore, it can further restrict what the user can see and act on. For example, you can use it to display only those objects with names that match a case-sensitive filter string. To change the object-level filter (page 62), use the **Edit Filter Details** window.

**Related Concepts**

“Appendix A. Features of the User Interface” on page 433
“Standard Windows” on page 434
“Session History” on page 441

**Related Tasks**

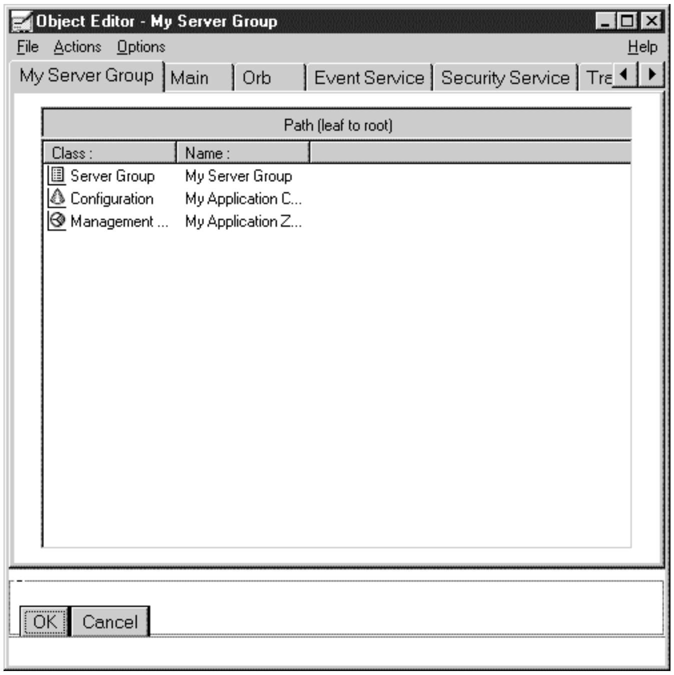
“Control Which Objects are Displayed” on page 62
“Chapter 2. Use the System Manager User Interface” on page 55

## The Object Editor Window

The Object Editor window can be used to display and edit the attributes of objects, including those that represent the CBConnector System Management components. It is a main tool for acting on objects through the CBConnector System Management user interface.

The Object Editor window, shown in Figure 75 on page 440, is displayed as a notebook with pages that group related attributes. One Object Editor window can be used to display and act on several objects.

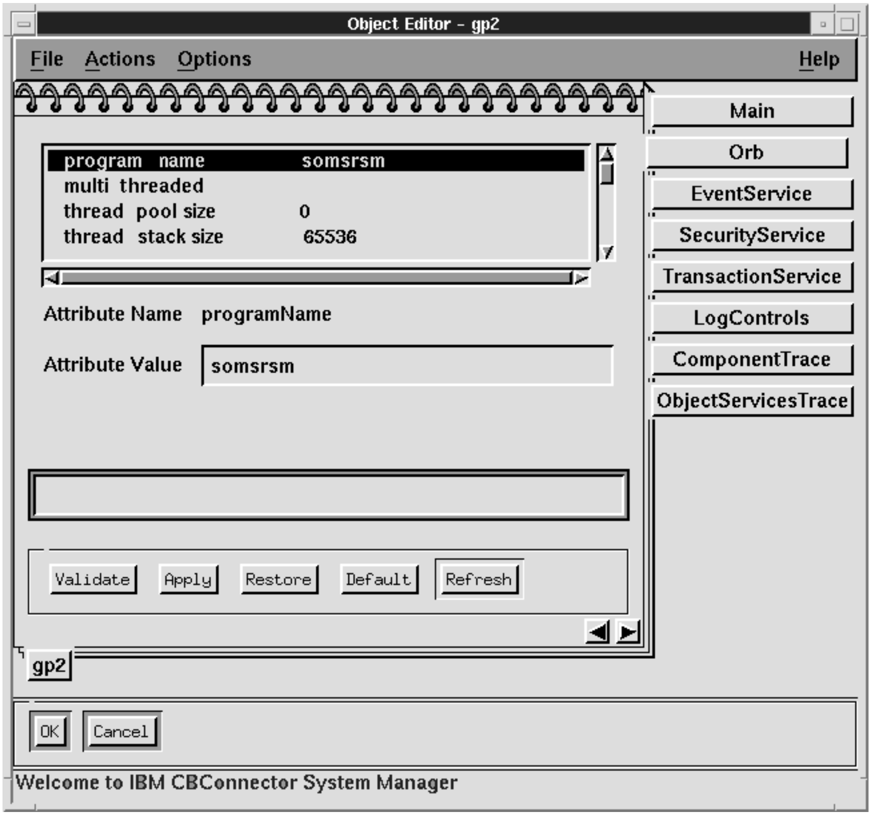
AIX The Object Editor window on AIX looks different to the window on Windows NT, but is functionally identical. For example, in both windows you click on a tab to display a notebook page of attributes, then can click on individual attributes to change them. For comparison, see Figure 76 on page 441.



Path (leaf to root)

| Class : | Name : |
|---|---|
| Server Group | My Server Group |
| Configuration | My Application C... |
| Management ... | My Application Z... |

*Figure 75. The Object Editor window for a Server Group model*



```
program  name          somsrsm
multi  threaded
thread  pool size        0
thread  stack size       65536
```

Attribute Name programName

Attribute Value somsrsm

Welcome to IBM CBConnector System Manager

Figure 76. The Object Editor window on AIX

### Display the Object Editor

You can display the Object Editor *after selecting at least one object* by any of the following methods:

- On the pop-up menu for the object, click **Edit**
- Select the **Selected - Edit** menu-bar choice

#### Related Concepts

“Appendix A. Features of the User Interface” on page 433
“Standard Windows” on page 434

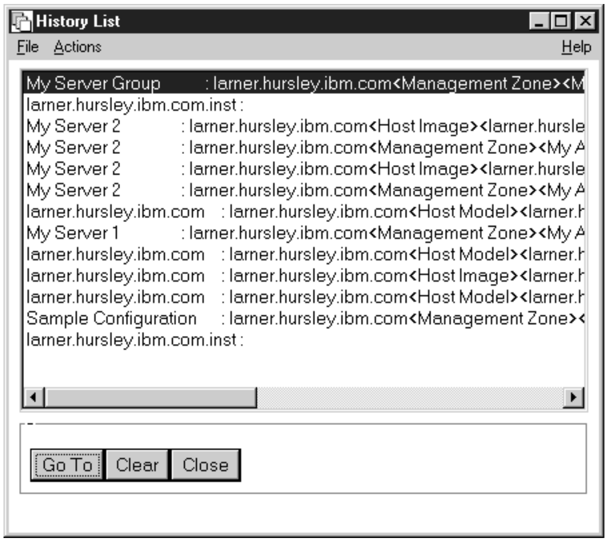
#### Related Tasks

“Edit Objects” on page 72
“Chapter 2. Use the System Manager User Interface” on page 55

## Session History

When you display an Information Controller window, you start a **session** that lasts until you close that window (or exit from the user interface). All objects that you open during a session are recorded in the **session history** for that window.

You can use the **History List** window, as shown in Figure 77, to display your session history and to quickly and easily revisit any object that was visited in the history list.



```
My Server Group        : larner.hursley.ibm.com<Management Zone><M
larner.hursley.ibm.com.inst :
My Server 2           : larner.hursley.ibm.com<Host Image><larner.hursle
My Server 2           : larner.hursley.ibm.com<Management Zone><My A
My Server 2           : larner.hursley.ibm.com<Host Image><larner.hursle
My Server 2           : larner.hursley.ibm.com<Management Zone><My A
larner.hursley.ibm.com   : larner.hursley.ibm.com<Host Model><larner.h
My Server 1           : larner.hursley.ibm.com<Management Zone><My A
larner.hursley.ibm.com   : larner.hursley.ibm.com<Host Model><larner.h
larner.hursley.ibm.com   : larner.hursley.ibm.com<Host Image><larner.h
larner.hursley.ibm.com   : larner.hursley.ibm.com<Host Model><larner.h
Sample Configuration     : larner.hursley.ibm.com<Management Zone><
larner.hursley.ibm.com.inst :
```

*Figure 77. The History List window*

To preserve direct access to an object between sessions, you can add the object to your **“Hotlist”**.

### Display the History List Window

You can display the History window by either of the following methods in the Information Controller window:

- Select the History button ⧉ of the Tool bar
- Select the **History** choice of the **Navigate** menu-bar choice

#### Related Concepts

“Appendix A. Features of the User Interface” on page 433
“Standard Windows” on page 434
“Hotlist”

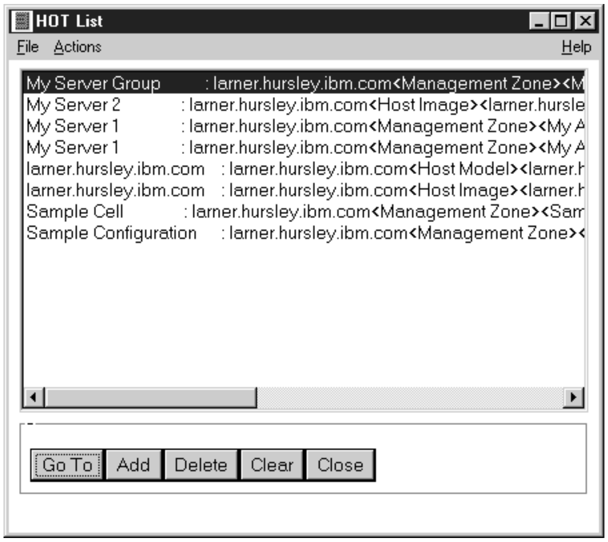
#### Related Tasks

“Chapter 2. Use the System Manager User Interface” on page 55
“Display Objects” on page 64

## Hotlist

Each user of the CBConnector System Management user interface can create a **hotlist** of significant objects that is preserved across sessions with the user interface. The hotlist forms a permanent list of objects that you can access directly during all sessions with the CBConnector System Management user interface.

You can use the **Hotlist** window, as shown in Figure 78, to display your hotlist and to quickly and easily open a selected object.



```
My Server Group        : larner.hursley.ibm.com<Management Zone><M
My Server 2          : larner.hursley.ibm.com<Host Image><larner.hursle
My Server 1          : larner.hursley.ibm.com<Management Zone><My A
My Server 1          : larner.hursley.ibm.com<Management Zone><My A
larner.hursley.ibm.com   : larner.hursley.ibm.com<Host Model><larner.h
larner.hursley.ibm.com   : larner.hursley.ibm.com<Host Image><larner.h
Sample Cell          : larner.hursley.ibm.com<Management Zone><Sam
Sample Configuration     : larner.hursley.ibm.com<Management Zone><
```

*Figure 78. The Hotlist window*

### Display the Hotlist Window

You can display the Hotlist window by either of the following methods in an Information Controller window:

- Select the **Hotlist** button ▦ of the Tool bar
- Select the **Hotlist** menu choice of the **Navigate** menu-bar choice.

**Related Concepts**

"Appendix A. Features of the User Interface" on page 433
"Standard Windows" on page 434
"Session History" on page 441

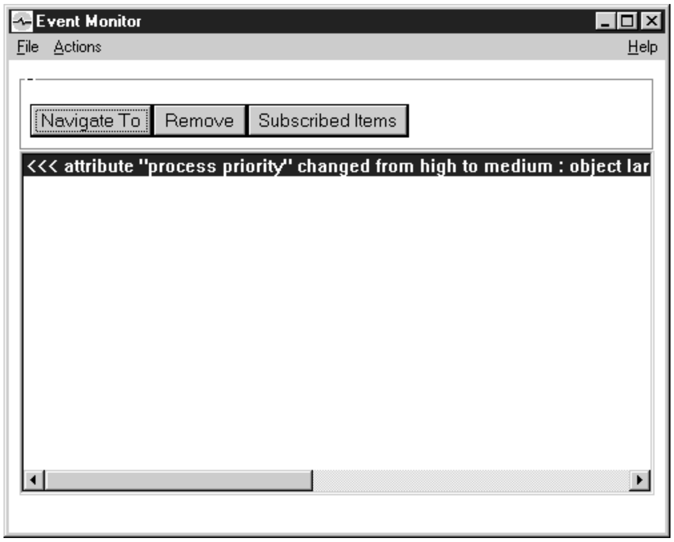
**Related Tasks**

"Chapter 2. Use the System Manager User Interface" on page 55
"Display Objects" on page 64

## Event Subscriptions

CBConnector System Management automatically records events that it has caused; for example, the starting of any of its asynchronous actions and the subsequent successful completion of that action. However, there are many more events that you can **subscribe** to; for example, any change in an object's attributes. You can use a set of **Event Monitor** windows to monitor and react to events.

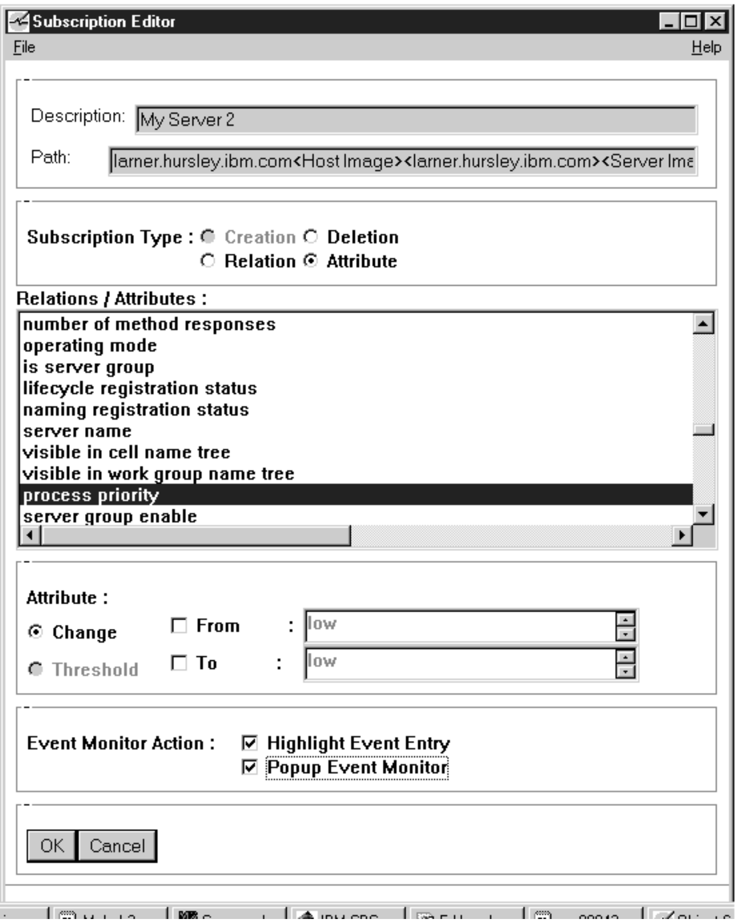
The Event Monitor window, shown in Figure 79 on page 444, is used to display events that have occurred, and to open the objects affected. It can also be used to display the other Event Monitor windows; for example, to change event subscriptions.



<<< attribute "process priority" changed from high to medium : object lar

*Figure 79. The Event Monitor window*

The Subscription Editor window for an object, shown in Figure 80 on page 445, is used to subscribe to new events, change existing event subscriptions, and delete existing event subscriptions. The Subscription Editor window acts on the one object for which it was displayed.



Description: My Server 2

Path: larner.hursley.ibm.com<Host Image><larner.hursley.ibm.com><Server Ima

Subscription Type : ○ Creation ○ Deletion
○ Relation ● Attribute

Relations / Attributes :

- number of method responses
- operating mode
- is server group
- lifecycle registration status
- naming registration status
- server name
- visible in cell name tree
- visible in work group name tree
- process priority
- server group enable

Attribute :

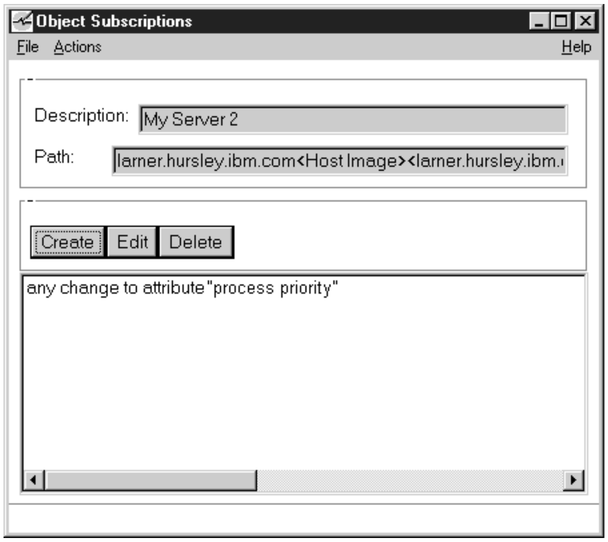
● Change □ From : low

○ Threshold □ To : low

Event Monitor Action : ☑ Highlight Event Entry
☑ Popup Event Monitor

*Figure 80. The Subscription Editor window*

The Object Subscriptions window for an object, shown in Figure 81 on page 446, is used to list the existing event subscriptions for that object. It can also be used to invoke the Subscription Editor window to create new subscriptions or to change or delete existing subscriptions.



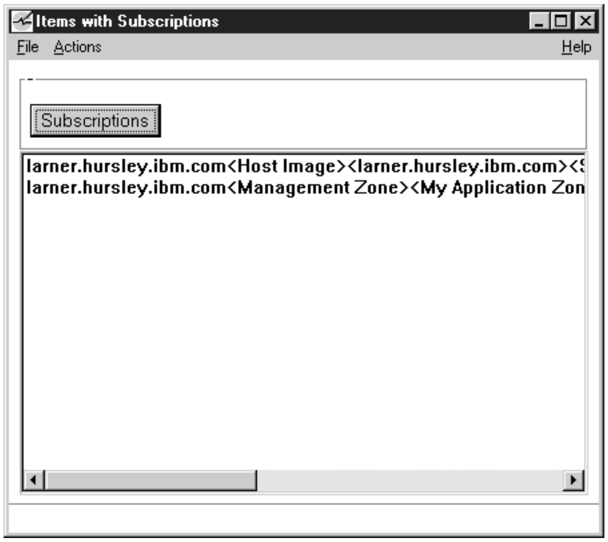
Description: My Server 2

Path: larner.hursley.ibm.com<Host Image><larner.hursley.ibm.

any change to attribute"process priority"

*Figure 81. The Object Subscriptions window*

The Items with Subscriptions window, shown in Figure 82, is used to list all the objects that have existing event subscriptions. It can also be used to invoke the Object Subscriptions window to display the event subscriptions for an object.



larner.hursley.ibm.com<Host Image><larner.hursley.ibm.com><

larner.hursley.ibm.com<Management Zone><My Application Zon

*Figure 82. The Items with Subscriptions window*

### Display the Event Monitor Windows

You can display the main Event Monitor window by either of the following methods in the Information Controller window:

- Select the Event Monitor icon of the Toolbar.

  or

- Select the **Event Monitor** menu choice from the **Events** pull-down menu.

To display the Items with Subscriptions window, do the following:

1. Display the Event Monitor window
2. Click the **Subscribed Items** action.

To display the Object Subscriptions window, do the following in either the Information Controller window or the Items with Subscriptions window:

1. Select an object
2. Select the **Subscriptions** action

**Related Concepts**

“Appendix A. Features of the User Interface” on page 433
“Standard Windows” on page 434
“Session History” on page 441
“Event Monitoring” on page 8

**Related Tasks**

Create or Change Event Subscriptions (page 420)
Delete Event Subscriptions (page 421)
“Chapter 2. Use the System Manager User Interface” on page 55

## Wizards

The System Manager user interface provides the following wizards to help you create and configure server groups or client styles as part of a system management Configuration.

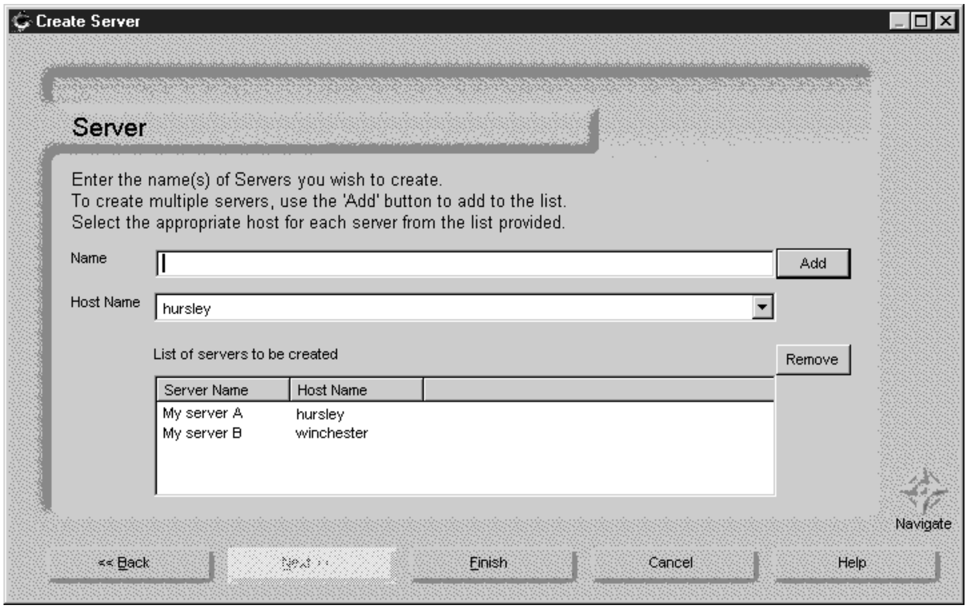
**Create Servers**
To create a server group, configure servers as members of the group onto their hosts, and optionally configure that group as a **controlled server group** for workload management.

**Create Clients**
To create a client style, configure it onto a host, and indicate the bootstrap host for the clients.

**Configure servers**
To configure applications onto server groups.

**Configure Clients**
To configure applications onto client styles.

Each wizard presents you with a series of panels, such as shown in Figure 83 on page 448, to prompt you for information needed, then completes the related task automatically.


Create Server

Server

Enter the name(s) of Servers you wish to create.
To create multiple servers, use the 'Add' button to add to the list.
Select the appropriate host for each server from the list provided.

Name

Host Name: hursley

List of servers to be created

| Server Name | Host Name |
|---|---|
| My server A | hursley |
| My server B | winchester |

*Figure 83. Example panel of a System Manager wizard*

Use of all these wizards follows the same simple procedure, outlined as follows. The only difference between the wizards is in the selections that you can make and the details that you can specify.

1. Start the wizard by selecting the appropriate menu option from the **Wizards** menu-bar choice of the System Manager user interface.
2. On the Management Zone panel, select the management zone within which you want to create or configure something. If this panel does not list an appropriate management zone, you can type the name of a new management zone, which will be created automatically by the wizard.
3. On the Configuration panel, select the Configuration (of the management zone that you selected previously) within which you want to create or configure something. If this panel does not list an appropriate Configuration, you can type the name of a new Configuration, which will be created automatically by the wizard.
4. On subsequent panels presented by the wizard, select options or type details as prompted on the panel.
5. To move from one panel to the next panel to be completed, click the **next** button.
6. To move from one panel to the previous panel completed, click the **back** button. This enables you to change details that you have specified earlier.
7. To move from one panel to any other panel, click the navigate icon.
8. To finish and have the wizard create or configure what you have selected, click the **finish** button on any panel. The wizard checks the details that you have specified, and if that is complete and suitable, the wizard performs the task.

   If you have not specified enough information or have specified wrong information it displays the panel that you need to use and prompts you for appropriate input. On any panel, select options or type details as prompted on the panel.
9. To leave the wizard without having it perform any action, click the **cancel** button on any panel.

**Note:** The wizard does not create or configure what you have selected until you click the **finish** button on any panel and have specified the details that it needs.

The servers or client styles created are added to your enterprise the next time that you activate the Configuration.

**Related Concepts**

"Appendix A. Features of the User Interface" on page 433

## Help Information

Each window of the System Manager user interface has general help information about the features of the window and the keys that you can use to act on it. You can also select context-sensitive help for areas of a window (for example, menu choices) and for the objects that are displayed in the window. Also, to learn what a Tool bar icon is for, you can rest your mouse pointer on the button for a few seconds to display a summary of the icon's function.

This enables you to keep on using the user interface to do the things you want as efficiently as possible, by providing *online* information specific to the window that you are using, the actions that you want to take, and the objects that you want to act on.

The help information provides links to other parts of the CBConnector System Management online information and, through that, to the rest of the Component Broker information web.

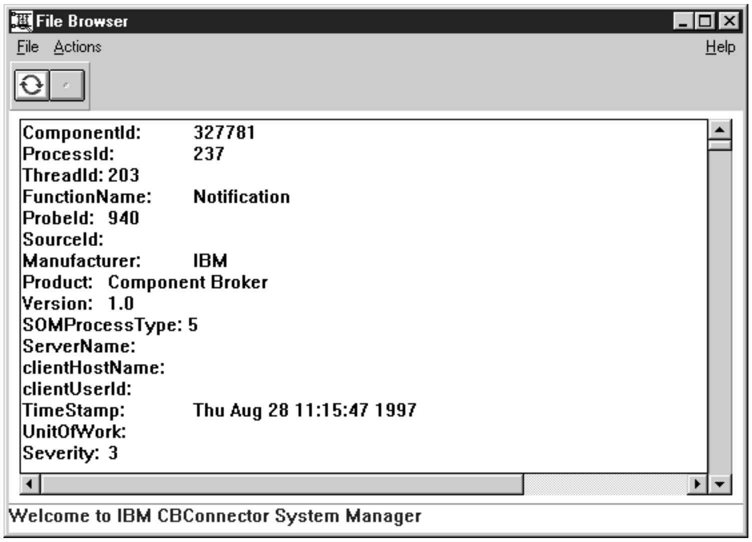
**Related Concepts**

"Appendix A. Features of the User Interface" on page 433
"Standard Windows" on page 434
"Session History" on page 441
"Chapter 2. Use the System Manager User Interface" on page 55
"The Component Broker Online Library" on page 474

## File Browser

The *file browser* can be used to display the contents of files on hosts managed by CBConnector System Management. In particular, it can display the activity, error, or trace log information for objects managed through the SM user interface. This helps administrators to identify any problems recorded in the log files. It provides a range of functions to help view the file contents; for example, it can be used to move around the file, change the format of the display, filter the display, and search for entries.

**The File Browser window**



```
ComponentId:        327781
ProcessId:          237
ThreadId: 203
FunctionName:       Notification
ProbeId:  940
SourceId:
Manufacturer:       IBM
Product:  Component Broker
Version:  1.0
SOMProcessType: 5
ServerName:
clientHostName:
clientUserId:
TimeStamp:          Thu Aug 28 11:15:47 1997
UnitOfWork:
Severity: 3
```

Welcome to IBM CBConnector System Manager

**Related Concepts**

“Appendix A. Features of the User Interface” on page 433
“Standard Windows” on page 434
Activity log (page 415)
Error log (page 415)
Trace log (page 416)

**Related Tasks**

“Display Information in Component Broker Logs” on page 423
“Chapter 2. Use the System Manager User Interface” on page 55

## Action Console Windows

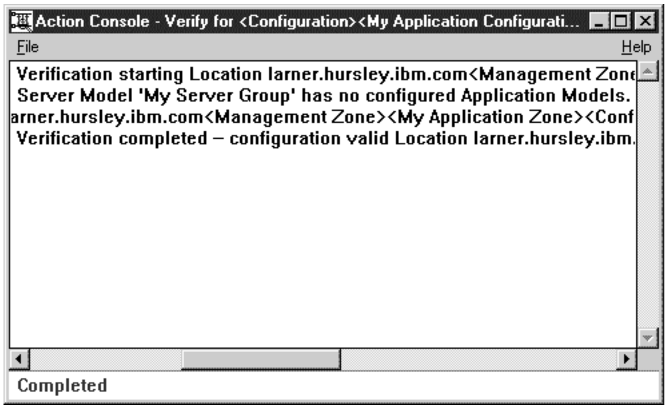
Some actions on objects through the System Manager user interface run asynchronously and can take more than an instant to complete; for example, an **Activate** action on a Configuration. For such actions, the SM user interface displays an *action console window* (as shown in Figure 84 on page 451), to notify you of the existence of the ongoing action and to provide progress messages.

While the action is in progress, **Running** is displayed on the status bar of the console window.

When the action has finished successfully, **Completed** is displayed on the status bar of the console window. The window can then be closed, if no longer required.

If the action fails to complete successfully, the console window provides messages about the errors that caused the failure. Some messages can refer you to look in the activity, error, and trace logs for more information, useful in resolving problems. You can use the file browser to display log files.

Action Console windows are not displayed for most actions, because they run synchronously and complete in a very short time.



Action Console - Verify for <Configuration><My Application Configurati...
File
Help

```
Verification starting Location larner.hursley.ibm.com<Management Zone
Server Model 'My Server Group' has no configured Application Models.
arner.hursley.ibm.com<Management Zone><My Application Zone><Conf
Verification completed – configuration valid Location larner.hursley.ibm.
```

Completed

*Figure 84. A Sample Action Console Window*

### Related Concepts

"Appendix A. Features of the User Interface" on page 433

# Appendix B. Quality of Protection

The Component Broker security service protects messages that flow between the client and the server. This is referred to as *qualities of protection* (QOP). In general, both DCE and SSL-based security can be used to protect these messages, however there are some differences that are worth noting. Component Broker defines six qualities of protection:

**Establish Trust in Client**
This ensure the requesting principal (either at a client process, or when a server invokes requests to other servers) is authenticated on any requests sent to a server.

**Establish Trust in Server**
This ensures the targeted server is authenticated on any requests sent to the server.

**Replay Detection**
This verifies that the same message is not sent twice. For example, this is useful to prevent someone from forcing a bank withdrawal request to occur twice by copying the first one.

**Out-of-Sequence Detection**
This verifies that two requests are not received out of order. For example, this is useful to prevent someone from forcing a bank withdrawal request from being processed before the deposit request; forcing the account to be overdrawn.

**Integrity**
This ensures the message content is not changed. For example, this is useful to prevent someone from changing the withdrawal amount. Integrity adds some overhead to method requests, because each message is signed at the client and verified at the server.

**Confidentiality**
This ensures that the message content can not be read. For example, this is useful to prevent someone from either seeing the withdrawal amount, or even the fact that this is a withdrawal request. Confidentiality adds a significant overhead to method requests, because each message is encrypted at the client and decrypted at the server.

All of these qualities are useful for guarding against cases where someone may be tapping a line between the client and server and watching message traffic flow by, and possibly attempting to change it.

The significant thing here is that SSL and DCE offer slightly different options, each use a different algorithm for performing the protection, and each form of protection has a different impact on overall performance. In particular, Component Broker's DCE-based security support does not allow you to select replay detection or out-of-sequence detection, even if integrity or confidentiality are turn on. DCE only supports mutual-authentication (*establish trust in client* and *establish trust in server*), whereas establish trust in client is optional with SSL.

DCE uses the Message Digest-5 (MD5) for integrity and Commercial Data Masking Facility (CDMF) for confidentiality. SSL uses Rivest, Shamir, Adleman (RSA) encryption for authentication, either MD5 or Secure Hash Algorithm (SHA) (in that order of preference) for integrity, and either Rivest Cipher-4 (RC4) or Rivest Cipher-2 (RC2) (in that order of preference) for confidentiality.

## Server Supports QOP and Server Requires QOP

When configuring your servers you can specify what QOP the server supports and what QOP the server requires of any clients that communicate with it. In this way, you can allow clients to communicate with a higher QOP than is actually required by the server. *A server always supports a QOP that is equal to or greater than the QOP it requires*.

## Client Performs QOP and Server (as client) Performs QOP

Where a server supports a higher QOP than it requires, the choice of what QOP to perform is left to the client. In some cases, even though a server requires a lower QOP, a client host may want to perform its requests with a higher level of protection; for example, if the client host is in a particularly vulnerable location. In this case, the client can be configured to perform the higher QOP.

In addition, the perform-QOP attributes for the client also govern whether communication with the server is allowed. If the server does not support the QOP that a client is configured to perform, then any requests from that client to that server are not allowed; they return to the application with a **NO_PERMISSION** system exception.

The actual QOP used between a client and server depends in part on the QOP attributes set for the client and the server, and the underlying security mechanisms used in the communication between a client and a server. This is also true of communication between servers and other servers, which is generally treated as a case of client to server communication. The resulting QOP between a client and server (or between server and server) is referred to as the *coalesced-QOP*. This is described further in “Enable Security within a Configuration” on page 321.

**Note:** Component Broker currently supports only DCE as the underlying security mechanism for server to server communication.

### Related Concepts

“Securing your Enterprise” on page 36
“Security” on page 263

### Related Tasks

“Configure Security for a Server” on page 323
“Configure Security for a Client Style” on page 326
“Configure Security for a Host Daemon” on page 329

### Related References

“Standard Export QOP Models for Servers” on page 455
“Standard Export QOP Models for Host Daemons” on page 457
“Standard Perform QOP Models for Servers” on page 458
“Standard Perform QOP Models for Client Styles” on page 460

## Standard Export QOP Models for Servers

The following table (page 455) lists the effect of the **Standard Export QOP Models** attribute, which can be set for servers to one of the following values: **authenticity**, **integrity**, and **confidentiality**. By default, this attribute is **authenticity**.

The effect of the **Standard Export QOP Models** attribute is filtered by the following other security attributes of the server, as shown in the table: **DCE Client Association Enabled** and **SSL Type-I Client Association Enabled**.

The actual quality of protection that is performed is determined by the QOP *exported* by the target server and the QOP to be *performed* by a given client (or server acting as a client to another server). To understand what quality of protection is actually performed, you can examine the Perform QOP Models attributes of the Client Styles and Servers:

- “Standard Perform QOP Models for Client Styles” on page 460
- “Standard Perform QOP Models for Servers” on page 458

For a description of each of the options set by the standard QOP models, see “Options Set by QOP Models for Servers” on page 464.

*Table 21.* **Standard Export QOP Models for Servers**

| Standard Export QOP Models | Export Client Authn Supt’d | Export Client Authn Req’d | Export Server Authn Supt’d | Export Server Authn Req’d | Export Message Replay Detect Supt’d | Export Message Replay Detect Req’d | Export Message OoS Detect Supt’d | Export Message OoS Detect Req’d | Export Message Integrity Supt’d | Export Message Integrity Req’d | Export Message Confident’l Supt’d | Export Message Confident’l Req’d |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Neither Security Association Enabled** | | | | | | | | | | | | |
| * | N | N | N | N | N | N | N | N | N | N | N | N |
| **DCE Security Association Enabled** | | | | | | | | | | | | |
| Authenticity | Y | Y | Y | Y | N | N | N | N | N | N | N | N |
| Integrity | Y | Y | Y | Y | N | N | N | N | Y | Y | N | N |
| Confidentiality | Y | Y | Y | Y | N | N | N | N | Y | Y | Y | Y |
| **SSL Type-I Association Enabled** | | | | | | | | | | | | |
| Authenticity | Y | Y | Y | Y | Y | Y | Y | Y | N | N | Y | N |
| Integrity | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | N |
| Confidentiality | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| **Both DCE and SSL Type-I Security Associations Enabled** | | | | | | | | | | | | |
| Authenticity | Y | Y | Y | Y | Y | Y | Y | Y | N | N | Y | N |
| Integrity | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | N |
| Confidentiality | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |

### Related Concepts

“Appendix B. Quality of Protection” on page 453
“Securing your Enterprise” on page 36
“Security” on page 263

### Related Tasks

“Configure Security for a Server” on page 323

### Related References

“Standard Perform QOP Models for Servers” on page 458
“Standard Perform QOP Models for Client Styles” on page 460

## Standard Export QOP Models for Host Daemons

The following table (page 457) lists the effect of the **Standard Export QOP Models** attribute, which can be set for host daemons to one of the following values: **authenticity**, **integrity**, and **confidentiality**. By default, this attribute is **authenticity**.

The **Standard Export QOP Models** attribute has effect only if the **security enabled** attribute is set to **yes** on the Daemon Image, as shown in the table.

The actual quality of protection that is performed is determined by the QOP *exported* by the daemon and the QOP to be *performed* by a given client. To understand what quality of protection is actually performed, you can examine the "Standard Perform QOP Models for Client Styles" on page 460.

The options set by the standard QOP models for host daemons are equivalent to those set by the standard QOP models for servers. For a description of each of the options set by the standard QOP models, see "Options Set by QOP Models for Servers" on page 464.

*Table 22.* **Standard Export QOP Models for Host Daemons**

| Standard Export QOP Models | Export Client Authn Supt’d | Export Client Authn Req’d | Export Server Authn Supt’d | Export Server Authn Req’d | Export Message Replay Detect Supt’d | Export Message Replay Detect Req’d | Export Message OoS Detect Supt’d | Export Message OoS Detect Req’d | Export Message Integrity Supt’d | Export Message Integrity Req’d | Export Message Confident’l Supt’d | Export Message Confident’l Req’d |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Security Disabled** | | | | | | | | | | | | |
| * | N | N | N | N | N | N | N | N | N | N | N | N |
| **(SSL) Security Enabled** | | | | | | | | | | | | |
| Authenticity | N | N | Y | Y | Y | Y | Y | Y | N | N | Y | N |
| Integrity | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | N |
| Confidentiality | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |

**Related Concepts**

“Appendix B. Quality of Protection” on page 453
“Securing your Enterprise” on page 36
“Security” on page 263

**Related Tasks**

Configure Security for a Host Daemon

**Related References**

“Standard Perform QOP Models for Client Styles” on page 460

## Standard Perform QOP Models for Servers

The following table lists the effect of the **Standard Perform QOP Models** attribute, which can be set for servers to one of the following values: **authenticity**, **integrity**, and **confidentiality**. By default, this attribute is **authenticity**.

The effect of the **Standard Perform QOP Models** attribute is filtered by the following security attributes of the target server, as shown in the table: **DCE Client Association Enabled** and **SSL Type-I Client Association Enabled**.

The actual quality of protection that is performed is determined by the QOP *exported* by the target server and the QOP to be *performed* by a given client (or server acting as a client to another server). This topic lists the quality of protection that is actually performed by a server for communication with a target server.

For a description of each of the options set by the standard QOP models, see “Options Set by QOP Models for Servers” on page 464.

*Table 23.* **Default Perform QOP Models for Servers**

| Standard Perform QOP Models | Perform Client Authent'n | Perform Server Authent'n | Perform Message Replay Detection | Perform Message OoS Detection | Perform Message Integrity | Perform Message Confident'l |
|---|---|---|---|---|---|---|
| **Neither Security Association Available to Target Server** | | | | | | |
| * | N | N | N | N | N | N |
| **DCE Security Association Available to Target Server** | | | | | | |
| Authenticity | Y | Y | N | N | N | N |
| Integrity | Y | Y | N | N | Y | N |
| Confidentiality | Y | Y | N | N | Y | Y |

### Related Concepts

"Appendix B. Quality of Protection" on page 453
"Securing your Enterprise" on page 36
"Security" on page 263

### Related Tasks

"Configure Security for a Server" on page 323

### Related References

"Standard Export QOP Models for Servers" on page 455
"Standard Perform QOP Models for Client Styles"
"Options Set by QOP Models for Servers" on page 464

## Standard Perform QOP Models for Client Styles

The following table lists the effect of the **Standard Perform QOP Models** attribute, which can be set for client styles to one of the following values: **authenticity**, **integrity**, and **confidentiality**. By default, this attribute is **authenticity**.

The effect of the **Standard Perform QOP Models** attribute is filtered by the following security attributes of the target server, as shown in the table: **DCE Client Association Enabled** and **SSL Type-I Client Association Enabled**.

The actual quality of protection that is performed is determined by the QOP *exported* by the target server and the QOP to be *performed* by a given client. This topic lists the quality of protection that is actually performed by a client for communication with a target server.

For a description of each of the options set by the standard QOP models, see “Options Set by QOP Models for Client Styles” on page 463.

*Table 24.* **Default Perform QOP Models for Client Styles**

| Standard Perform QOP Models | Perform Client Authent'n | Perform Server Authent'n | Perform Message Replay Detection | Perform Message OoS Detection | Perform Message Integrity | Perform Message Confident'l |
|---|---|---|---|---|---|---|
| **Neither Security Association Available to Target** | | | | | | |
| * | N | N | N | N | N | N |
| **DCE Security Association Available to Target Server** | | | | | | |
| Authenticity | Y | Y | N | N | N | N |
| Integrity | Y | Y | N | N | Y | N |
| Confidentiality | Y | Y | N | N | Y | Y |
| **SSL Association Available to Target Server** | | | | | | |
| Authenticity or Integrity | Y | Y | Y | Y | Y | N |
| Confidentiality | Y | Y | Y | Y | Y | Y |
| **Both DCE and SSL Associations Available to Target Server** | | | | | | |
| Authenticity | Y | Y | Y | Y | N | N |
| Integrity | Y | Y | Y | Y | Y | N |
| Confidentiality | Y | Y | Y | Y | Y | Y |

### Related Concepts

"Appendix B. Quality of Protection" on page 453
"Securing your Enterprise" on page 36
"Security" on page 263

### Related Tasks

"Configure Security for a Client Style" on page 326

### Related References

"Standard Export QOP Models for Servers" on page 455
"Standard Export QOP Models for Host Daemons" on page 457
"Standard Perform QOP Models for Servers" on page 458
"Options Set by QOP Models for Client Styles" on page 463

## Options Set by QOP Models for Client Styles

This topic describes the effect of the options set by the **Standard Perform QOP Models** attributes for client styles, as listed in the tables in the topic “Standard Perform QOP Models for Client Styles” on page 460. These options are set when you configure security for a client style, to specify a standard perform QOP model.

**perform client authentication**

Set this attribute to **yes** (the default) if the client will authenticate itself to any servers if given a choice. If the server requires client authentication, then this client will always authenticate itself. If the server *supports* client authentication, but does not *require* it, then this client will attempt to authenticate itself only if this attribute is set to **yes**. By default, this attribute is **yes**.

**perform server authentication**

Set this attribute to **yes** (the default) if servers must authenticate themselves to this client on method requests.

- If the server *requires* that it be authenticated with this client, then server authentication will always be performed, regardless of the setting on this attribute.
- If the server *supports* server authentication but *does not require* it, and this attribute is set to **yes** then the server will be authenticated.
- If the server *does not support* server authentication, and this attribute is set to **yes**, then requests will not be permitted to that server.

By default, this attribute is **yes**.

**perform message replay detection**

Set this attribute to **yes** if this client requires servers to perform message replay detection on method requests.

- If the server *requires* message replay detection then message replay detection is always performed, regardless of the setting on this attribute.
- If the server *supports* message replay detection but *does not require* it, and this attribute is set to **yes**, then message replay detection is requested on method requests to that server.
- If the server *does not support* message replay detection, and this attribute is set to **yes**, then requests are not permitted to that server.

By default, this attribute is **yes**.

**perform message out-of-sequence detection**

Set this attribute to **yes** if the client requires message out-of-sequence detection on method requests.

- If the server *requires* message out-of-sequence detection then message out-of-sequence detection is always performed, regardless of the setting on this attribute.
- If the server *supports* message out-of-sequence detection but *does not require* it, and this attribute is set to **yes**, then message out-of-sequence detection is requested on method requests to that server.
- If the server *does not support* message out-of-sequence detection, and this attribute is set to **yes**, then requests are not permitted to that server.

By default, this attribute is **yes**.

**perform message integrity**

Set this attribute to **yes** (the default) if the client requires message integrity protection on method requests.

- If the server *requires* message integrity, then message integrity will always be performed, regardless of the setting on this attribute.
- If the server *supports* message integrity but *does not require* it, and this attribute is set to **yes**, then message integrity is requested on method requests to that server.
- If the server *does not support* message integrity, and this attribute is set to **yes**, then requests are not permitted to that server.

By default, this attribute is **yes**.

**perform message confidentiality**

Set this attribute to **yes** (the default) if the server requires message confidentiality protection on cascaded method requests.

- If the server *requires* message confidentiality, then message integrity will always be performed, regardless of the setting on this attribute.
- If the server *supports* message confidentiality but *does not require* it, and this attribute is set to **yes**, then message confidentiality is requested on method requests to that server.
- If the server *does not support* message confidentiality, and this attribute is set to **yes**, then requests are not permitted to that server.

### Related Concepts

“Appendix B. Quality of Protection” on page 453
“Securing your Enterprise” on page 36
“Security” on page 263

### Related Tasks

“Configure Security for a Client Style” on page 326

### Related References

“Standard Perform QOP Models for Client Styles” on page 460

## Options Set by QOP Models for Servers

This topic describes the effect of the options set by the **Standard Perform QOP Models** attributes for servers, as listed in the tables in the topic “Standard Perform QOP Models for Servers” on page 458. These options are set when you configure security for a server to specify a standard perform QOP model.

**export client authentication supported**

Set this attribute to **yes** (the default) if clients are able to authenticate themselves with the server. This determines what is indicated in the *supported associations QOP* included in exported IORs. The same choice is included in both DCE and SSL tagged components. By default, this attribute is **yes**. If the **Security Enabled** attribute is set to **yes**, this attribute must also be set to **yes**.

**export client authentication required**

Set this attribute to **yes** (the default) if clients *are required to* to authenticate themselves with the server. This determines what is indicated in the *required associations QOP* included in exported IORs. The same choice is included in both DCE and SSL tagged components. By default, this attribute is **yes**. However, if **Export Client Authentication supported** is set to **yes**, then this attribute must also be set to **yes**.

**export server authentication supported**
Set this attribute to **yes** (the default) if clients are able to authenticate the server. This determines what is indicated in the *supported associations QOP* included in exported IORs. The same choice is included in both DCE and SSL tagged components. By default, this attribute is **yes**. If the **Security Enabled** attribute is set to **yes**, this attribute must also be set to **yes**.

**export server authentication required**
Set this attribute to **yes** (the default) if clients *are required to* authenticate the server. This determines what is indicated in the *required associations QOP* included in exported IORs. The same choice is included in both DCE and SSL tagged components. By default this attribute is **yes**. However, if **export server Authentication supported** is set to **yes**, this attribute must also be set to **yes**.

**export message confidentiality supported**
Set this attribute to **yes** if clients are able to request message confidentiality protection. This determines what is indicated in the *supported associations QOP* included in exported IORs. This attribute is set to **yes** if **SSL Type-I Client Association Enabled** is set to **yes**, otherwise it is set to **no** (the default).

**export message confidentiality required**
Set this attribute to **yes** if clients must request message confidentiality protection. This determines what is indicated in the *required associations QOP* included in exported IORs. The same choice is included in both DCE and SSL tagged components. By default this attribute is **yes**. This attribute is set to **yes** if **SSL Type-I Client Association Enabled** is set to **yes**, otherwise it is set to **no** (the default).

**export message integrity supported**
Set this attribute to **yes** if clients are able to request message integrity protection. This determines what is indicated in the *supported associations QOP* included in exported IORs. This attribute is set to **yes** if **SSL Type-I Client Association Enabled** is set to **yes**, otherwise it is set to **no** (the default).

**export message integrity required**
Set this attribute to **yes** if clients must request message integrity protection. This determines what is indicated in the *required associations QOP* included in exported IORs. This attribute is set to **yes** if **SSL Type-I Client Association Enabled** is set to **yes**, otherwise it is set to **no** (the default).

**export message out-of-sequence detection supported**
Set this attribute to **yes** if clients are able to request out-of-sequence detection. This determines what is indicated in the *supported associations QOP* included in exported IORs. This attribute is set to **yes** if **SSL Type-I Client Association Enabled** is set to **yes**, otherwise it is set to **no** (the default).

**export message out-of-sequence detection required**
Set this attribute to **yes** if clients *must request* out-of-sequence detection. This determines what is indicated in the *required associations QOP*

included in exported IORs. This attribute is set to **yes** if **SSL Type-I Client Association Enabled** is set to **yes**, otherwise it is set to **no** (the default).

**export message replay detection supported**
Set this attribute to **yes** if clients are able to request replay detection. This determines what is indicated in the *supported associations QOP* included in exported IORs. This attribute is set to **yes** if **SSL Type-I Client Association Enabled** is set to **yes**, otherwise it is set to **no** (the default).

**export message replay detection required**
Set this attribute to **yes** if clients *must request* replay detection. This determines what is indicated in the *required associations QOP* included in exported IORs. This attribute is set to **yes** if **SSL Type-I Client Association Enabled** is set to **yes**, otherwise it is set to **no** (the default).

**perform client authentication**
Set this attribute to **yes** (the default) if the server will authenticate itself to any downstream servers if given a choice. If the downstream server requires client authentication, then this server always authenticates itself. If the downstream server *supports* client authentication, but does not *require* it, then this server attempts to authenticate itself only if this attribute is set to **yes**.

**perform server authentication**
Set this attribute to **yes** (the default) if the server requires any downstream servers to authenticate themselves to this server on cascaded method requests.

- If the downstream server requires that it be authenticated with this server then downstream server authentication is always performed, regardless of the setting on this attribute.
- If the downstream server *supports* server authentication but does not *require* it, and this attribute is set to **yes**, then the downstream server is authenticated.
- If the downstream server does not support server authentication, and this attribute is set to **yes**, then requests are not permitted to the downstream server.

**perform message replay detection**
Set this attribute to **yes** if the server requires message replay detection on cascaded method requests.

- If the downstream server requires message replay detection, then message replay detection is always performed, regardless of the setting on this attribute.
- If the downstream server *supports* message replay detection but does not *require* it, and this attribute is set to **yes**, then message replay detection is requested on the cascaded method request to that server.
- If the downstream server does not support message replay detection, and this attribute is set to **yes**, then requests are be permitted to that downstream server.

By default, this attribute is **no**.

**perform message out-of-sequence detection**
Set this attribute to **yes** if the server requires message out-of-sequence detection on cascaded method requests.

- If the downstream server requires message out-of-sequence detection, then message out-of-sequence detection is always performed, regardless of the setting on this attribute.
- If the downstream server *supports* message out-of-sequence detection but does not require it, and this attribute is set to **yes**, then message out-of-sequence detection is requested on the cascaded method request to that server.
- If the downstream server does not support message out-of-sequence detection, and this attribute is set to **yes** then requests are not permitted to that downstream server.

By default, this attribute is **no**.

**perform message integrity**

Set this attribute to **yes** (the default) if the server requires message integrity protection on cascaded method requests.

- If the downstream server requires message integrity, then message integrity protection is always performed, regardless of the setting on this attribute.
- If the downstream server *supports* message integrity but does not *require* it, and this attribute is set to **yes**, then message integrity is requested on the cascaded method request to that server.
- If the downstream server does not support message integrity, and this attribute is set to **yes**, then requests are not permitted to that downstream server.

**perform message confidentiality**

Set this attribute to **yes** (the default) if the server requires message confidentiality protection on cascaded method requests.

- If the downstream server requires message confidentiality, then message integrity is always performed, regardless of the setting on this attribute.
- If the downstream server *supports* message confidentiality but does not *require* it, and this attribute is set to **yes**,then message confidentiality is requested on the cascaded method request to that server.
- If the downstream server does not support message confidentiality, and this attribute is set to **yes**, then requests are not permitted to that downstream server.

### Related Concepts

“Appendix B. Quality of Protection” on page 453
“Securing your Enterprise” on page 36
“Security” on page 263

### Related Tasks

“Configure Security for a Server” on page 323

### Related References

“Standard Perform QOP Models for Servers” on page 458
“Standard Perform QOP Models for Client Styles” on page 460

# Appendix C. Related Topics

The following topics are part of the *System Administration Guide*, because they provide information that is related to system management, but perhaps not used during the normal administration phase of Component Broker. For example, tasks that use the DDL Editor, such as “Configure an Extension to the Quality of Service for Managed Objects”, are normally completed before the application affected is packaged.

- “Configure an Extension to the Quality of Service for Managed Objects”
- “Configure DB2 to use the CBConnector Data Cache Facility” on page 470

## Configure an Extension to the Quality of Service for Managed Objects

Use this procedure to configure an extension to the quality of service for an application’s managed objects. If you do not do this task, the quality of service of a managed object is defined by the Business Object Instance Manager (BOIM) application adapter.

You need do this task only if instructed by an application adapter developer.

This task creates and configures an Application Adapter Formation Install object within the application’s DDL file.

This task description uses the DDL Editor to edit the application’s DDL file, which is named after the application family, and resides in the application family’s /bin directory.

Before starting this task, you should be familiar with using the DDL Editor to edit DDL files. (For more information about the DDL Editor, see the online Component Broker information library.

**To configure an extension to the quality of service for managed objects, complete the following steps using the DDL Editor:**

1. Display the DDL Editor with your DDL file; for example, by typing the following command at a commandline window:

   ```
   somsmddle ddl_filename
   ```

   where, *ddl_filename* is the name of the ddl file to be edited.
2. From the pop-up menu of your Application Family Install, click **New - Application Adapter Formation**. This displays a dialog box window for you to name the new Application Adapter Formation.
3. In the dialog box window, type the name for the new Application Adapter Formation, then click **OK**.
4. To display the new Application Adapter Formation, expand the **Application Adapter Formation Installs** folder.
5. From the pop-up menu of your Application Adapter Formation Install, click **Edit**. This displays the Object Editor notebook.
6. Change the following attributes of the Application Adapter Formation, as instructed by the application adapter developer:

**install type**
: Whether the application adapter **`extend`**s the BOIM quality of service or **`replace`**s that service; default, **`extend`**. For this release, this attribute should always be set to **`extend`**.

**create function name**
: The name of the function used to create the application adapter's configuration object

**description**
: An optional 256-character text field for you to store any text

**integer1 ... integer4**
: Numeric configuration parameters needed to support the additional behavior of the application adapter. The value of each attribute can be an integer equal to or greater than 0.

**string1 ... string4**
: ASCII configuration parameters needed to support the additional behavior of the application adapter. The value of each attribute can be up to 256 ASCII characters.

7. To apply the changes and close the Object Editor window, click the **OK** button.
8. From the pop-up menu of your Application Adapter Formation Install, click **Drag**.
9. Expand the **Application Installs** folder, to display the Application that is to use the new Application Adapter Formation.
10. From the pop-up menu of the Application Install, click **Configure Application Adapter Formation**. This creates a Provides Application Adapter Formation relationship between the Application and the Application Adapter Formation.
11. From the pop-up menu of your Application Family Install, click **generate source**. This starts an asynchronous action to update the source in the application's DDL file, and displays an action console window that you can use to monitor the progress of the action. If any errors are encountered, messages are displayed in the action console window. Also, when the action has completed, it displays a **Completed** message on the status bar at the bottom of the action console window.

You can now continue to configure the application in the normal manner. The extended quality of service will be implemented when you next activate a Configuration that contains models for the application and the Servers it is configured to run on. This normally follows several tasks to add the application into a Configuration, configure it onto servers, then activate the Configuration.

## Configure DB2 to use the CBConnector Data Cache Facility

Bind the files db2cntrr.bnd and db2cntcs.bnd to each of your databases that will be accessed from your Component Broker applications. To do this issue the following DB2 commands:

```
connect to
<your>
bind drive:\directory\db2cntrr.bnd  datetime iso  grant public  blocking all  isolation rs
bind drive:\directory\db2cntcs.bnd  datetime iso  grant public  blocking all
```

If you want the cache facility to use db2 repeatable read locks, (not recommended) then issue the command

```
bind db2cntrr.bnd  datetime iso grant public blocking all isolation rr
```

These files create 2 packages in the database. The package names are ″userid.db2cntrr″ and ″userid.db2cntcs″.

Make sure that the userid used by the CBConnector server to connect to the database has the following authority:

1. authority to connect to the database
2. authority to execute the packages userid.db2cntrr and userid.db2cntcs
3. authority to read and update your application tables

Update your db2 dbms configuration to use the required TP_MON_NAME, as follows:

- WIN **TP_MON_NAME=somtrx1i**. Use the db2 command:

  ```
  Update dbm cfg using tp_mon_name somtrx1i
  ```

- AIX **TP_MON_NAME=libsomtrx1.so**. Use the db2 command:

  ```
  Update dbm cfg using tp_mon_name libsomtrx1.so
  ```

## Problem Determination

Component Broker provides many runtime information sources to help you perform problem determination. The information sources primarily consist of messages issued through System Manager user interface and entries added to the activity and error logs.

If such sources do not enable you to completely resolve a problem, you can turn on tracing of Component Broker services to provide detailed records of process flows. This should normally only be done under advice from your IBM representative.

A basic description of information that might be useful for preliminary problem determination is provided in the topic “Chapter 15. Operate your Enterprise” on page 413.

Detailed descriptions of the information and procedures for problem determination is provided in the *Problem Determination Guide*.

### Related Concepts

“Sources of Information” on page 415

### Related Tasks

“Control Component Broker Trace” on page 426

## Introduction to SNA

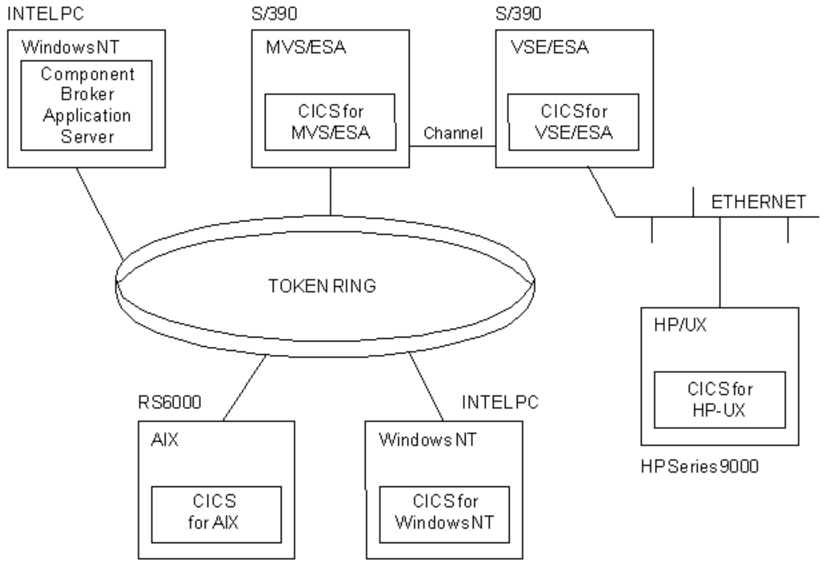
This topic provides an introduction to the concepts and terminology of Systems Network Architecture (SNA) that you need to be familiar with when using Component Broker in an SNA network. When preparing for Component Broker applications servers to communicate with other systems using APPC, it is important that you have an understanding of the SNA products that the other systems use. This is because although the two systems must agree common parameters, the terminology used for these parameters may be different.

IBM’s Systems Network Architecture (SNA) defines a set of rules that systems use to communicate. These rules define the layout of the data that flows between the systems, and the action that the systems take when they receive the data. The data layout and actions are known as the *formats* and *protocols*, and together they constitute the architecture.

SNA does not specify how a system should implement the rules. Indeed a fundamental objective of SNA is to allow systems that have very different internal hardware and software design to be able to communicate. The only requirement is that the externals meet the rules of the architecture.

The figure An Example Heterogeneous SNA Network (page 472) illustrates a heterogeneous network, with a Component Broker application server intercommunicating with CICS regions on a number of different host platforms.

**An Example Heterogeneous SNA Network**



Each CICS region uses its own platform-specific SNA product. CICS/400 uses OS/400 Intercommunication Facility (ICF). IBM mainframe-based CICS works very closely with Virtual Telecommunications Access Method (VTAM) to support SNA.

There are many ways to connect systems in an SNA network, and provided that the data is successfully transferred in the correct format, the systems are unaware of the make-up of the network. To reflect this, SNA configuration is done at two levels, the *logical level*, and the *physical level*:

- **The logical level** is the level concerned with the characteristics of the systems that wish to communicate.

  *Logical Unit (LU)* is an SNA term used to describe a logical collection of services that can be accessed from the network. In an SNA network, you can think of a Component Broker application server as an LU. SNA defines many different types of LU, including devices like terminals and printers. The type of LU that is used by Component Broker application servers for APPC communication is *LU type 6.2*.

  Each LU is identified by a name of up to eight characters, referred to as the *LU name*. An SNA network also has a name of up to eight characters, called the *network name*. The network name is sometimes referred to as the *network id* or the *netid*. An LU can be uniquely identified by combining its LU name with the network name of the network that owns it. The LU's name is then referred to as the *network-qualified LU name*, or the *fully-qualified LU name*. For example, if an LU named **CBSERV1** belonged to a network named **MYSNANET**, then its network-qualified LU name would be **MYSNANET.CBSERV1**.

For an LU to communicate with another LU, it must establish at least one *session* between them. The request to activate a session is referred to as a *bind request*. It is used to pass details of the capabilities of the initiating LU to the receiving system, and also to determine a route though the network. The receiving LU is then given a chance to send a description of its capabilities to the initiating LU in the *bind response*. Once the session is established, it may be used for a number of APPC requests and remains active for as long as the two LUs, and the network between them, is available.

When you configure your network, you can set up different characteristics for the sessions that are established between a pair of LUs; for example, the route they take through the network. Session characteristics are defined in what are referred to as *modegroups*, where all the sessions associated with a modegroup have the same characteristics. Modegroups are identified by a *modename* of up to eight characters.

- **The physical level** is concerned with linking the actual host machines, or *nodes*, in the network. Each node has physical links, or *connections*, to other nodes so that they are all connected to at least one other node. Data may have to travel along a number of links in order to get from one system to another. Also, these links may be of different types; for example:
  - IBM Token Ring
  - Synchronous Data Link Control (SDLC)
  - Ethernet
  - X.25

  These types of links are collectively referred to as *data link control (DLC) protocols*.

  Each node has a *physical unit (PU)*. This is a combination of hardware and software that controls the links to other nodes. There are a number of types of PU that reflect the capabilities and responsibilities of the PU; for example:

**PU type 5**
The best known example is an IBM mainframe processor running VTAM. VTAM provides the support for the Systems Services Control Point (SSCP) function defined in SNA.

**PU type 4**
This is a communications controller, such as ACF/NCP, that resides in the center of a network, routing and controlling the data flow between machines.

**PU type 2**
This is a small machine, such as an APPC workstation. It can only communicate directly with a PU type 4 or a PU type 5 and relies on it to route the data to the correct system.

**PU type 2.1** This is a more advanced PU type 2 that can also communicate with other PU type 2.1 nodes directly. This node is capable of supporting an *independent LU*. An independent LU is an LU that can establish a session with another LU without using VTAM. Communications Server (used by Component Broker application servers) is a PU type 2.1 node.

PU type 2.1 nodes may have support for *Advanced Peer-to-Peer Networking (APPN)*. This enables a node to search for an LU in the network rather than requiring a remote LU's location to be preconfigured locally. There are two types of APPN nodes, *end nodes* and *network nodes*. An end node is able to receive a search request for an LU and respond indicating whether the LU is local to the node or not. A network node is able to issue search requests, as well as respond to them, and maintains a dynamic database that contains the results of the search requests the node has made. Support for APPN can greatly reduce the maintenance work in an SNA network, especially if the network is large or dynamic. IBM Communications Server supports APPN.

**Related Concepts**

"IBM Communications Server" on page 34
"Communicating across SNA connections" in the *CICS Intercommunication Guide*
*Systems Network Architecture Technical Overview* provides a comprehensive summary of SNA, and its Bibliography contains a range of SNA publications.

**Related Tasks**

"Collect information for your SNA configuration" on page 358
"Configure Communications Server" on page 367
"Configure an APPC Connection to a Tier-3 System for use by Applications" on page 354

## The Component Broker Online Library

The Component Broker online Information Library tries to help you accomplish what you need to do when you need to do it.

The library provides you with four main information categories of information:

**Tasks** guide you through step-by-step instructions

**Concepts**
explain what you should understand in order to accomplish certain tasks

**Reference**
provides detailed programming specifications

**Programming Guides**
describe what is involved in the design and implementation of Component Broker applications

In addition, the library provides a Glossary to explain key terms, and a powerful, full-text Search engine.

The online library is in HTML format, so if you have browsed the World Wide Web, you already know how to use it effectively.

Find related information quickly. In Concepts and Tasks, links to Related Topics are located at the end of each page. The Programming Guides and Reference are structured in book format. At the end of these pages, you can find links to the Table of Contents for that book, as well as links to next and previous pages.

From the online library you can print information as either individual html topics or complete pdf versions of the Component Broke bo oks. (All Component Broker hardcopy information is provided in the form of softcopy books that you can view and print using the Adobe Acrobat reader.)

For more information about the Component Broker online library and help information, see the **Help for Help** icon and information on the library interface.

See also “Component Broker Information” on page xv.

# Appendix D. Notices

This publication 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
500 Columbus Avenue
Thornwood, NY 10594
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-chrome, 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 publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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 information and all licensed material available for it are provided by IBM under terms of the 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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written.

These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

## Trademarks

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

AIX
CICS
DB2
IBM
MVS/ESA
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.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

VeriSign is exclusively licensed to VeriSign, Inc.

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

# Bibliography

This topic lists books related to system management of Component Broker that are available separate from Component Broker.

Many books related to system management of Component Broker (for example, about communications with tier-3 systems) are available on the World Wide Web. For example,

- http://www.ibmlink.ibm.com/, IBMLink, which provides information about IBM manuals.
- http://www.software.ibm.com/ts/txseries/library/manuals/nt/, the bookshelf for IBM Transaction Server for Windows NT, Version 4. This bookshelf contains an online version of the *Using IBM Communications Server for Windows NT with CICS*.
- http://www.software.ibm.com/ts/cics/library/manuals/, the bookshelves for IBM CICS Clients and Gateways books.

## IBM Communications Server for Windows NT books

- *Using IBM Communications Server for Windows NT with CICS*, SC33-1900
- *Communications Servers for Windows NT Up and Running Guide*, GC31-8424
- *Client/Server Communications Programming*, SC31-8425
- *System Management Programming*, SC31-8426

## SNA books

- *Systems Network Architecture Technical Overview*, GC30-3073
- *Systems Network Architecture Transaction Programmer's Reference Manual for LU Type 6.2*, GC30-3084
- *Systems Network Architecture—Sessions Between Logical Units*, GC20-1868

- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*, SC30-3269
- *Systems Network Architecture LU 6.2 Reference—Peer Protocols*, SC31-6808

## CICS books

- *Intercommunication Guide*, SC33-1564
- *Administration Guide*, SC33-1774
- *Administration Reference*, SC33-1563
- *Application Programming Reference*, SC33-1569
- *CICS Family: Interproduct Communication*, SC33-0824
- *Using Microsoft SNA Server Version 2 with CICS*, SC33-1899
- *Using IBM Communications Server for AIX with CICS*, SC33-1898
- *CICS for MVS/ESA Intercommunication Guide*, SC33-1695
- *CICS Family: Communicating from CICS on System/390*, SC33-1697
- *CICS/VSE Intercommunication Guide*, SC33-0701
- *CICS for OS/2 Intercommunication*, SC33-1583
- *CICS/400 Intercommunication*, SC33-1388

# Index

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC09-2704-03