IBM® Distributed Computing Environment Version 3.2 for AIX®

Quick Beginnings

IBM® Distributed Computing Environment Version 3.2 for AIX®

Quick Beginnings

Note
Before using this document, read the general information under "Appendix D. Notices" on page 135.

First Edition (July 2001)

This edition applies to Version 3.2 of *IBM Distributed Computing Environment for AIX* and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Order publications through your IBM representative or the IBM branch office that serves your locality. IBM does not stock publications at the address below.

IBM welcomes your comments. Send your comments to the following address:
International Business Machines Corporation
Department VLXA
11400 Burnet Road
Austin, Texas
78758

Contents

Tables v

Welcome to DCE 3.2 for AIX vii
Typographic and Keying Conventions . . . vii

Part 1. Understanding DCE 3.2 for AIX. 1

Chapter 1. Overview of DCE 3.2 for AIX . . 3
What Is DCE? 3
Comparing Legacy DCE Security Registry and LDAP Integrated DCE Security Registry . . . 4
Product Contents. 5
DCE 3.2 Licensed Programs 5
DCE Version 3.2 Base Services for AIX . . 5
DCE Version 3.2 for AIX 9
IBM Enhancements to DCE 10
Standards Conformance 13
DCE compatibility with AIX 13
Unsupported OSF DCE Features 16
Limitations of Supported Services for Accounts Configured to use Public Key Authentication 18
Limitations of Supported Services for LDAP Security Servers. 18

Part 2. Planning for, Installing, Uninstalling, and Migrating to DCE 3.2 for AIX 21

Chapter 2. Planning 23
System Requirements 23
Disk Space Requirements 24
Global and Cell Considerations 24
Planning Questions to Consider 24
Establishing a Cell Name 27
The Cell Namespace 29
Planning for Access Control. 34
DCE Naming Considerations for Internationalization 35
Client and Server Considerations 36
Determining Requirements for DCE Client Machines 36
Determining Requirements for DCE Server Machines 40
Determining Requirements for Legacy DCE Migration to an LDAP Security Server 44
DCE Administration Utilities 44
Application Development Environment . . . 46
Location of Installed DCE Files. 46
The /opt/dcelocal Subtree 47
Conventional UNIX Directories 47
File Locations 47

Chapter 3. Installing, Uninstalling, and Migrating to DCE 3.2 for AIX 51
Installable Packages 51
Requisite Software 54
Installing DCE 3.2 58
Software Processes to Stop 58
Running the Easy Installation Program . . 58
Special Installation Instructions 60
Migrating an AIX DCE Cell to DCE 3.2 for AIX 61
Migrating an Existing DCE Cell to Exploit DCE Security Registry and LDAP Integration . 69
Uninstalling DCE 3.2 70
Suggested Reading. 70

Part 3. Configuring, Starting, and Stopping DCE 3.2 for AIX 71

Chapter 4. Configuring DCE 3.2 for AIX Servers and Clients 73
Configuring DCE 73
Overview of Configuration 73
User-Supplied Commands 77
Environment Variables 80
Initial Cell Configuration. 80
Configuring Servers 81
Configuring Clients 85
Further Cell Configuration 94
Configuring DTS Servers. 94
Configuring a DTS Client 97
Configuring Secondary CDS Servers . . . 97
Configuring Security Replica Servers. . . 99

Configuring the Global Directory Agent 101
Configuring EMS Servers 102
Configuring SNMP Servers 103
Configuring DCE 3.2 for AIX Security Integration 103
Configuring Audit Servers 104
Configuring Password Strength Servers 105
Configuring the Name Service Interface Daemon (NSID) 107
Configuring an Identity Mapping Server 108
Configuring DCE Web Secure for AIX . . . 109
Configuring DCE Web Secure Using SMIT 109
Configuring DCE Web Secure from a Command Line 110
Verifying Configuration of DCE Web Secure. 110
Unconfiguring DCE Components 112
Considerations Before Unconfiguring . . 113
Split Unconfiguration 114
Steps for Unconfiguring DCE 115
Unconfiguring DCE Web Secure 116

Chapter 5. Starting and Stopping DCE 3.2 for AIX 119
Starting DCE Daemons 119
Using the Command Line to Start Daemons. 119
Using SMIT to Start DCE Immediately and at System Restart 120
Changing Arguments Passed to DCE Daemons at Restart 120
Stopping DCE Daemons 121

Chapter 6. Obtaining Additional Information 123
Books 123
Online Information 123
Online Books 123
Help Files 124
Print and Order Books 124
IBM DCE Publications 124
Other Publications 124
Using DCE 3.2 for AIX Documentation . . 124
The start_dcedoc program 124
Viewing Plain Text Files. 125
Viewing the HTML Documentation using a Web Browser. 125
Printing the PDF Books 126

Appendix A. Online Documentation . . . 127

Appendix B. DCE Web Secure for AIX Advanced Configuration 131
Authenticated Path Configuration 131
Keyfile Configuration 131
Summary of Advanced Configuration Syntax 132
Examples of Advanced Configuration . . . 132

Appendix C. Dummy Filesets 133

Appendix D. Notices 135
Trademarks 137

Index 139

Tables

1. Installation filesets and requisite software 54
2. DCE Software Bundles 59

Welcome to DCE 3.2 for AIX

This book describes the IBM® Distributed Computing Environment Version 3.2 for AIX® (DCE 3.2 for AIX) product. It includes information for both server and client components.

"Part 1. Understanding DCE 3.2 for AIX" on page 1 gives an overview of DCE 3.2 for AIX.

"Part 2. Planning for, Installing, Uninstalling, and Migrating to DCE 3.2 for AIX" on page 21 explains how to plan for, install, uninstall, and migrate to DCE 3.2 for AIX.

"Part 3. Configuring, Starting, and Stopping DCE 3.2 for AIX" on page 71 explains how to configure and use DCE 3.2 for AIX.

Typographic and Keying Conventions

This guide uses the following typographic conventions:

Bold **Bold** words or characters represent system elements that you must use literally, such as commands, options, and pathnames.

Italic *Italic* words or characters represent variable values that you must supply. *Italic* type may also introduce a new DCE term.

`Constant width`
Examples and information that the system displays appear in `constant width` typeface.

[ ] Brackets enclose optional items in syntax descriptions and format.

{ } Braces enclose a list from which you must choose an item in syntax descriptions and format.

| A vertical bar separates items in a list of choices.

< > Angle brackets enclose the name of a key on the keyboard. Angle brackets also enclose the descriptive name of a variable.

... Horizontal ellipsis points indicate that you can repeat the preceding item one or more times.

This guide uses the following keying conventions:

<Ctrl- *x***> or ^** *x*
The notation **<Ctrl-** *x***>** or **^** *x* followed by the name of a key indicates

a control character sequence. For example, **<Ctrl-C>** means that you hold down the control key while pressing **<C>**.

<Return>
The notation **<Return>** refers to the key on your terminal that is labeled with the word Return or Enter, or with a left arrow.

Part 1. Understanding DCE 3.2 for AIX

Chapter 1. Overview of DCE 3.2 for AIX

IBM Distributed Computing Environment Version 3.2 for AIX (DCE 3.2 for AIX) is a member of the IBM Server Series family of products. DCE 3.2 for AIX is based on Open Software Foundation (OSF) DCE technology (Release 1.2.2).

What Is DCE?

DCE provides a standard environment that supports distributed applications. It represents technologies that are selected by the OSF and has emerged as the industry standard for distributed services.

An application written to use DCE runs in any environment that supports the OSF DCE standard. DCE makes it possible for application developers to give users secure access to the wide range of information and services available within their network. DCE does this while hiding the complexity of the network environment.

Distributed computing services, as implemented in DCE, provide an important enabling software technology for the development of distributed applications. DCE makes the underlying network architecture transparent to application developers. It consists of a software layer between the operating system and network interface and the distributed application program. DCE provides a variety of common services that users need for development of distributed applications. These applications include name and time services, and a standard remote procedure call (RPC) interface. DCE provides a means for application developers to design, develop, and deploy distributed applications.

A group of DCE machines that work together and that DCE administers as a unit is called a *cell*. For example, imagine an organization comprised of several departments, each in a different building, and each operating on its own budget. Each department in such an organization could have its own DCE cell.

A DCE environment is a group of one or more DCE cells that can communicate with each other. A cell becomes a part of a DCE environment when it obtains access to one or more global directory services in which the other cells in the environment are registered.

If two cells for two different departments are a part of a DCE environment, then a user in one department's cell can access resources in another

department's cell. This access is typically less frequent and more restricted than access to resources within the user's own cell.

An administrator can configure a DCE cell in many ways, depending on its users' requirements. A cell consists of a network connecting two kinds of nodes:

- **DCE user (client) machines** are general-purpose DCE machines. They contain software that enables them to act as clients to all of the DCE services.
- **DCE server machines** include special software enabling them to provide one or more of the DCE services. Every cell must have at least one of the following servers in order to function:
  - Cell Directory Server
  - Security Server

Other DCE servers can be present in a given DCE cell to provide additional functionality. For example a Global Directory Agent can enable the cell's directory server to communicate with other cells' directory servers.

DCE 3.2 for AIX is a layer between the AIX operating system, network services, and a distributed application. DCE 3.2 for AIX provides the services that allow a distributed application to interact with a collection of possibly heterogeneous computers, operating systems, and networks as if they were a single system. DCE 3.2 for AIX includes a set of standard services, software interfaces, and tools that support the creation, use, and maintenance of distributed applications in a diverse computing environment.

DCE 3.2 for AIX has the same organization as OSF DCE. Part 1 of this book introduces the concept of a DCE cell and gives a brief summary of how different machines participating in a Distributed Computing Environment are organized.

DCE 3.2 for AIX is based on the OSF DCE Release 1.2.2 code base and designed for the supported versions of the AIX operating system. See the *IBM DCE Version 3.2 for AIX: Release Notes* for a listing of the supported versions of the AIX operating system.

Comparing Legacy DCE Security Registry and LDAP Integrated DCE Security Registry

A feature introduced with DCE 3.2 for AIX enables users to integrate their existing Security Registry with LDAP such that all registry data can be stored within an LDAP directory.

For the purpose of this documentation, references to *legacy DCE* refer to any version of DCE that does not exploit the use of this feature, including DCE 3.2.

Product Contents

DCE 3.2 for AIX is available in the following Licensed Programs:

- **DCE Version 3.2 Base Services for AIX** which includes the following packages:
  - **DCE Client Services, Version 3.2**
  - **DCE System Management Interface Tool (SMIT), Version 3.2**
  - **DCE System Management, Version 3.2**
  - **DCE X.500 API Library, Version 3.2**
  - **DCE Tools for Application Developers, Version 3.2**
  - **DCE Messages, Version 3.2**
  - **DCE Online Documentation, Version 3.2**
  - **DCE Privacy Level Protection (previously known as the DCE Data Encryption Standard (DES) Library), Version 3.2**
- **DCE Version 3.2 for AIX** which includes all the packages found in DCE Version 3.2 Base Services for AIX as well as the following packages:
  - **DCE Cell Directory Server, Version 3.2**
  - **DCE Security Server, Version 3.2**

DCE 3.2 Licensed Programs

This section provides a description of the DCE 3.2 Licensed Programs and packages.

DCE Version 3.2 Base Services for AIX

The DCE Version 3.2 Base Services for AIX Licensed Program provides support for remote procedure calls, the client functionality for cell directory service, security, time, messaging and serviceability. This package also provides support for integrating DCE security services with AIX base operating system security. Because the DCE architecture is built on a threads-based model, the DCE Base requires a reentrant (threadsafe) version of the AIX C library, **libc_r.a**. The AIX operating system includes the **libc_r.a** library. DCE administration includes tools for configuring a cell, adding and deleting users in a cell, adding servers and clients to a cell, and similar functions.

Client Services

The following client services are provided as part of the DCE Version 3.2 Base Services for AIX:

- The **Remote Procedure Call (RPC)** facility enables you to create and run client applications and server applications. The RPC runtime service implements the network protocols by which the client side and the server side of an application communicate.
- **DCE Threads Compatibility Library for AIX** provides a programming model for building concurrent applications that perform many operations simultaneously. It provides support for multithreaded applications (that are based on POSIX 1003.4a Draft 4) that use the DCE threading model. The AIX package includes the DCE Threads Compatibility Library for AIX.
- **Multithreaded Programming Environment** support allows multiple threads to call standard C library functions without interfering with one another.
- **Distributed Time Service (DTS)** provides synchronized time in the distributed network environment on the computers that participate in a Distributed Computing Environment. DTS synchronizes a DCE host's time with Coordinated Universal Time (UTC), an international time standard.
- The **Cell Directory Service (CDS) client** provides the interface, **cdsclerk**, between CDS client applications and CDS servers. The **slim client** also provides the **cdsclerk** interface and most of the functionality of the regular client, but because **dced** is not executing on a slim client machine, no endpoint processing can occur.
  - **CDS Preferencing** enables administrators to specify a preferenced CDS clearinghouse from which a client will obtain CDS information. This feature is provided to improve performance at CDS clients by enabling cell administrators the ability to specify a preferred CDS clearinghouse from which a client will obtain CDS information. This is useful in situations where, for example, a low-performance WAN connects multiple high-performance LANs, and each of the LANs contains a CDS replica clearinghouse. With this feature, administrators can specify local clearinghouses as preferred over distant clearinghouses, and then clients use the distant clearinghouses only when the local clearinghouses are unable to satisfy requests.
- The **Security client** provides the following services:
  - **AIX Security Integration** coordinates the AIX base operating system security services with the DCE security services. This allows a user to log in to AIX and obtain DCE credentials at the same time. For more information about AIX Security Integration, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.
  - **GSSAPI Extensions** are a set of application program interfaces (APIs) that provide non-RPC applications the ability to use the DCE security authentication protocol. The system administrator can use GSSAPI to establish credentials or extract Extended Privilege Attribute Certificates (EPACs) for a non-RPC application.

  - **Extended Registry Attributes** (ERAs) expand the static registry attributes of Principal, Group, and Account to a dynamic set of registry attributes that can be customized to a cell.
  - The **Audit Service** performs the logging of audit records based on specified criteria. The Audit Service has three basic components:
    - **Application Programming Interfaces (APIs)** provide the functions that are used to detect and record critical events when the server services a client. They are also used to create tools that examine and analyze the audit event records.
    - The **Audit Daemon** maintains the filters and the audit logs.
    - **Audit Management Interfaces** allow you to specify how the Audit Daemon will filter the recording of Audit Events. These interfaces are available from the DCE Control Program (**dcecp**).
  - The **Enhanced Password Strength Server** extends the capabilities of the password strength server in previous DCE releases. The enhanced server allows you to control the following characteristics of user passwords:
    - Password composition
    - Password age
    - Password history and re-use
    - Password dictionaries and user-defined rules
- **DCE Web Secure** provides DCE credentials to CGI programs. DCE Web Secure must be installed and configured on a workstation that has a DCE client and an iPlanet FastTrack 4.1, iPlanet Enterprise 4.0, or iPlanet Enterprise 4.1 Web server. DCE Web Secure also supports the Web servers supported by DCE 3.1. DCE Web Secure for AIX can be configured using SMIT.

DCE System Management Interface Tool (SMIT)

The System Management Interface Tool (SMIT) for DCE allows you to perform DCE system management tasks using SMIT, **smitty** (a TTY version of SMIT), or DCE commands. You must install the DCE SMIT options before you can access the SMIT menus for DCE. SMIT uses interactive menus to guide users through many system management tasks.

DCE System Management

DCE System Management provides three management tools: DCE Event Management Service (EMS), the DCE Simple Network Management Protocol (SNMP) SubAgent, and DCE Web Administration.

- **EMS** provides asynchronous event support for DCE based applications. DCE EMS manages event services in a DCE cell. EMS consists of two parts — the **emsd** (EMS daemon) server and APIs to access event services through an interface to the suppliers, consumers, and event service administration for use by EMS clients. For more information about EMS, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

- **SNMP Subagent** provides network management support in the TCP/IP environment for monitoring DCE resources and services. System administrators and system management application programmers can use SNMP to monitor the DCE environment. This allows them to focus on making their resources and services more manageable. For more information about SNMP, see the *IBM DCE Version 3.2 for AIX and Solaris: Application Development Guide—Core Components*.
- **DCE Web Administration** provides a CGI program that administrators can use to manage DCE users, groups, organizations, and permissions. The CGI program is accessible from your Web browser at:

  `http://<servername>/dceweb`

  where <*servername*> is the name of the server where you installed and configured DCE Web Secure. This fileset requires installation and configuration of the DCE Web Secure fileset.

DCE X.500 API Library

The DCE X.500 API Library provides application programming interfaces to the CDS namespace. A library of functions is available with which to access the Directory Services.

DCE Tools for Application Developers

DCE Tools for Application Developers (dce.tools) includes the following tools for DCE administrative and application development support:

- The Interface Definition Language tool consists of a language (and its compiler) that supports the development of distributed applications following the client and server model. It automatically generates code that transforms procedure calls into network messages.
- The **sams** compiler generates files that can be used to add messaging and serviceability support to DCE applications.

DCE Messages

The Messages package includes versions of menus and error messages translated into languages other than English.

Online Documentation

The Online Documentation for DCE 3.2 for AIX provides IBM documentation sets in the following formats:

- HTML
- PDF
- Plain text

For more information concerning this documentation, refer to "Chapter 6. Obtaining Additional Information" on page 123.

DCE Privacy Level Protection

DCE Privacy Level Protection (**dce.priv.rte**), previously known as the Data Encryption Standard (DES) Library, provides a programming interface that enables remote procedure call (RPC) application data encryption. The **dce.priv.rte** feature utilizes the Data Encryption Standard (DES) algorithms that are part of the DCE Base Services for AIX. This feature includes support for User Data Masking Encryption, which was formerly packaged separately.

DCE Version 3.2 for AIX

The DCE Version 3.2 for AIX Licensed Program includes all the packages found in DCE Version 3.2 Base Services for AIX as well as the following packages:

DCE Cell Directory Server

The DCE Cell Directory Server is a central repository for information about resources in the distributed system. Typical resources are users, machines, and RPC-based services. The information consists of the name of the resource and its associated attributes. Typical attributes include a user's home directory or the location of an RPC-based server.

- The Directory Service consists of the **Cell Directory Service** (CDS) and the **Global Directory Agent** (GDA). The CDS manages a database of information about the resources in a group of machines called a DCE cell. The CDS also provides location-independent naming for servers. The GDA enables intercell communications by locating cells which have been registered in the global naming environment.
- **GDA Integration with LDAP** is an extension to GDA that allows the resolution of non-Domain Name System (DNS) style foreign cell names. With X.500 directories and any directories that support the LDAP protocol, the administrator can establish intercell communication. For more information about LDAP, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

DCE Security Server for AIX

DCE Security Server for AIX enables secure communications and controlled access to resources. It provides a set of the following security-related functions:

- The **Authentication Service** enables two processes on different machines to be certain of each other's identity.
- **Secure Communication** integrates DCE Remote Procedure Call (RPC) with the Security Service to protect communication.
- **Authorization** controls access to resources by comparing the credentials conferred to a user by the Privilege Service with the rights to the resource. The resource's Access Control List specifies rights to the resource.
- **Privilege Server** makes the following checks once identity has been established: Is the user authorized to access a resource? What permissions

are required? Does the user have those permissions? Use of Authenticated RPC generally invokes authentication and authorization for the user.

- The **Access Control List Facility** lists users who are authorized to access a given resource. An ACL API allows programmers to manipulate ACLs. The **dcecp acl** commands or the **acl_edit** command allow users to modify ACLs associated with resources that they own. These commands also allow users to modify to whom (user or group) access is granted and what specific permissions are given.
- The **Login Facility** initializes a user's DCE security environment by authenticating the user to the Security Service by means of the user's password. The Login facility then returns security credentials that authenticate the user to the required distributed services.
- **Public Key certificate login** allows a user to obtain initial DCE credentials by using an X.509v3 digital certificate and its associated public key pair to prove the user's identity. This feature is an extension of the OSF DCE 1.2.2 public key login protocol based on OSF RFC 68.4 (draft 7). This implementation requires the Entrust public key infrastructure (PKI).
- **Security Replication** enables the Master Registry Database to be replicated to one or more subordinate Registry Databases. The dcecp **registry** commands or the **sec_admin** command are the interfaces used to view and manipulate the state of both master and subordinate replicas.
- The **Identity Mapping Service** is used by the Authentication Service to determine a DCE user's identity when the user logs in with the public key certificate authentication protocol. The Identity Mapping Service maps a user's name in a public key certificate to a DCE principal name.
- **Security Registry and LDAP Integration** moves the legacy Security Registry to an LDAP directory. This enables users who have LDAP in their environment to further exploit LDAP by enabling them to move their DCE users to an LDAP directory. All existing interfaces are supported.

IBM Enhancements to DCE

The following services and commands contained in the previously listed DCE 3.2 Licensed Programs are IBM enhancements and extensions to the AIX implementation of DCE:

- **Added Services:**
  - **User Data Masking Encryption Facility**
  - **Enhanced Password Strength Server**
  - **CDS Preferencing**
  - **DCE Web Secure**
  - **Simple Network Management Protocol (SNMP)**
  - **Event Management Service (EMS)**

- **AIX Security Integration**
- **Public Key Certificate Login**
- **GDA Integration with LDAP**
- **Identity Mapping Server (IDMS)**
- **Slim Client**
- **DCE Security Registry and LDAP Integration**
- **GSSAPI Enhancements**
- **Entrust 5.0 Recertification**

• **Additional Commands**:

– Configuration Commands:

Notes:

1. The following commands use a different command format than the config command format used in AIX DCE V2.1, however, DCE 3.2 still supports the config command format used previously.
2. The following commands are not compatible with the dcecp **host configure**, **host start**, **host stop**, and **host unconfigure** commands.

chpesite
Updates the **pe_site** file, which contains the addresses of the security servers that you use.

clean_up.dce
Cleans up recreatable database files, cache files, and credential files. Use this command if problems arise when trying to start DCE.

config.dce
Configures and starts DCE components. This command provides for a split configuration of clients. Administrative configuration and local configuration can be performed separately. See "Further Cell Configuration" on page 94 for more information.

kerberos.dce
Creates the host principals, FTP principals, and key table entries used to support the Secure Remote commands.

migrate.dce
Migrates DCE configuration data from previous releases for use with the current release. There is no need to reconfigure when installing a new release of DCE. See "Migrating an AIX DCE Cell to DCE 3.2 for AIX" on page 61 for more information.

mkdceweb
Configures DCE Web Secure and DCE Web Administration into a Netscape FastTrack or Netscape Enterprise Web server or an iPlanet FastTrack or iPlanet Enterprise Web server.

mkreg.dce
Adds information about a DCE cell into the DOMAIN namespace.

rmdceweb
Unconfigures DCE Web Secure and DCE Web Administration from a Netscape FastTrack or Netscape Enterprise Web server or an iPlanet FastTrack or iPlanet Enterprise Web server.

rmreg.dce
Removes information about a DCE cell from the DOMAIN namespace.

show.cfg
Displays the local host's DCE or DFS™ configuration or both configurations. The **dce** and **dfs** option allows display of DCE information

start.dce
Starts the configured DCE components. This command makes sure that all components start in the correct order.

stop.dce
Stops the configured DCE components. This command makes sure that all components stop in the correct order.

unconfig.dce
Removes configuration of DCE components. This command provides for a split unconfiguration, with which administrative unconfiguration and local unconfiguration can be performed separately. See "Further Cell Configuration" on page 94 for more information.

– CDS Commands:

cdsdel Deletes recursively the namespace of a cell.

cdsli Lists recursively the namespace of a cell.

– RPC Commands:

rpcprotseqs
Determines the supported protocol on a given host.

rpcresolve
Recursively resolves the elements of a namespace entry.

– Security Commands:

rmxcred
Purges expired tickets from the credentials directory.

Standards Conformance

- DCE 3.2 for AIX supports the standards listed below, but cannot claim conformance to these standards because some of them are not in final form or because conformance tests do not exist.

  Threads
  - POSIX 1003.4a, draft 4
  - AES/Distributed Computing - Threads

  RPC AES/Distributed Computing - Remote Procedure Call

  Security
  - Authentication
    - Kerberos Version 5, draft 4
  - Authorization
    - POSIX 1003.6, draft 12 (acls)
  - AES/Distributed Computing - Security
  - GSSAPI, including Internet RFC 1964

  Directory
  - AES/Distributed Computing - Directory Services
  - X/OPEN-X.400 API Association XDS API Draft 6

  Transport Glue
  - RFC 1006, TPO-to-TCP

  Time
  - RFC 1129, NTP
  - AES/Distributed Computing - Directory Services

DCE compatibility with AIX

This section describes the compatibility of DCE for AIX with the supported versions of AIX for the RISC System/6000®.

- The AIX **man** command is *not* supported to display current DCE reference documentation. However, **dceman**, which displays a single manual page for DCE commands and subroutines, is supplied and emulates the AIX **man** command. When the documentation package is installed, **dceman** is linked to the **/usr/bin** directory, along with **asciiview** and **start_dcedoc**. To invoke the **dceman** command, type:

  ```
  dceman -lang locale dce_command
  ```

  where *dce_command* can be any command or subroutine documented in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* or *IBM DCE Version 3.2 for AIX and Solaris: Application Development Reference*. The –lang parameter enables the user to view DCE commands or subroutines in any language. *locale* can be substituted with one of the following locale names:

Notes:

1. Not all books are translated for each language. The commands and subroutines documented in non-translated books will view in English.
2. Any value specified for the –lang option that is not a valid locale will default to displaying the English documentation for the *dce_command*.

en_US English locale

ja_JP Japanese (EUC) locale

Ja_JP Japanese (PC Kanji) locale

JA_JP Japanese (UTF-8) locale

ko_KR
Korean locale

KO_KR
Korean (UTF-8) locale

pt_BR Brazilian Portuguese locale

PT_BR
Brazilian Portuguese (UTF-8) locale

zh_TW
Simplified Chinese (EUC) locale

Zh_TW
Simplified Chinese (GBK) locale

ZH_TW
Simplified Chinese (UTF-8) locale

- AIX Network Computing System (NCS) version 1.5.1 and the DCE Base Services for AIX can coexist on the same system because the DCE **dced** process provides the functionality that NCS applications expect from the **llbd** command.
- The AIX base operating security services have been integrated with the **DCE Security Services**. This integration presents the typical end-user with a single-system image rather than separate images of a local UNIX® system and a remote DCE system. There are some limitations to the integration, explained fully in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*; however, most users will be able to acquire DCE credentials through AIX commands (like **login** and **su**), will be able to change their DCE passwords through the AIX **passwd** command, and will be able to retrieve UNIX-type information from the DCE registry through the standard AIX **libc.a** routines, **getpwnam()**, **getpwuid()**, **getgrnam()**, and **getgrgid()**. The more general routines, **getpwent()** and **getgrent()**, are not DCE-aware at this time.

- **Important Note on Changing Passwords**: AIX Password operations are directed toward the registry defined by the **registry** user attribute or, in the absence of a **registry** attribute definition, to the registry defined by the **AUTHSTATE** environment variable.

  Password operations are not directed to both local and DCE registries.

  Changing passwords for a DCE-only user is done through **dcecp**. Changing passwords (both DCE and local) for a synchronized user (a user defined both locally and in DCE) can be done with the AIX **passwd** command in a 2-step procedure:

  ```
  $ AUTHSTATE=DCE passwd
  $ AUTHSTATE=compat passwd
  ```

  Passwords must be kept synchronized for synchronized users, or else either DCE or local authentication will fail. Also, if a user exists locally on more than one machine, the local password must be synchronized on all machines.

  The nonintegrated DCE security commands like **dce_login** (for logging onto DCE) and **dcecp** for DCE password-changing and registry queries are still available. Of course, local access is prerequisite to using these commands.

  The local administrator must set up the system to enable DCE for AIX security integration; complete configuration instructions can be found in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide*.
- **DCE ACLs** differ from AIX ACLs. DCE has its own commands, **acl_edit** and **dcecp**, to manipulate DCE ACLs on objects in the DCE namespace. The AIX commands (**acledit**, **aclget**, and **aclput**) do not work on objects in the DCE namespace, including files and directories in the DCE DFS filespace. AIX commands that back up and restore data do not maintain DCE ACLs on DCE LFS directories and files. Use the DCE DFS backup facility to back up and restore DCE LFS filesets.
- The AIX **dbx** debugging command has the capability to recognize and debug multiple threads. For more information on the debugger, see the *IBM DCE Version 3.2 for AIX and Solaris: Application Development Guide—Core Components*.
- The following discusses C++ and DCE compatibility.
  - Using **C++** with DCE requires a few considerations, but generally nothing beyond what is required in using a C based library with C++. The primary factor is using the correct version of **xlC**. DCE requires at least **xlC** version 3.1.3 in order to link with the DCE libraries.

    When DCE is installed, it will create a link **xlC_r4** to **xlC**. This should be used when compiling and linking DCE applications. This name determines the stanza in the **/etc/xlC.cfg** file that is used to control the

compiler configuration. Using the wrong version or the wrong linked name of the compiler can lead to problems at compile, link, and runtime.

- DCE exceptions are separate from the exceptions provided by the C++ language specification. The primary limitation in using DCE exceptions within C++ programs is that when a DCE exception is raised, destructors will not be called as the stack is unwound. The programmer must make sure that the objects are freed explicitly when DCE exceptions are handled. This might eliminate the use of automatically–allocated objects within segments of the application code.
- As with any C functions called from C++, be sure to include DCE header files in external C declarations. This makes sure that the C++ linkage looks for the non-mangled C names, not C++ names.

  In C, memory is typically allocated using **malloc**. In C++, memory is allocated using new *object_type*. DCE adds **rpc_ss_allocate** for volatile data that needs to be freed by the system after an rpc returns. Care needs to be taken to make sure that memory allocated by one method is always freed using the corresponding routine.

  As with any C library used in C++, it can be difficult to maintain a *pure* object-oriented architecture. In many cases, the components in DCE are fairly object-oriented in design. However, because most of the pieces of DCE are designed to work together, they often pass data structures between mostly unrelated functions. For example, a login handle is an opaque data type that has a core of several closely related functions to manage and maintain the login context. While this lends itself well to grouping the data and functions as an object, the handle will need to be passed either implicitly or explicitly to most other objects that might be created. Since it is bad form to expose a data value inside an object, a sophisticated design needs to be considered (possibly a handle or surrogate object).
- AIX 4.3.x does not support Cset 3.1.4. If you are compiling a C++ application on AIX 4.3.x, you need to use the IBM C and C++ compiler for AIX v3.6, which supports AIX 4.1.4, 4.2.x, and 4.3.x.
- Applications built using the AIX VACPP 5.0 compiler are supported by DCE 3.2 running on the 32–bit kernel of AIX 5.1.

Unsupported OSF DCE Features

The differences are grouped into sections by type. Each section is further subdivided into functional categories, which correspond with specific DCE services (such as configuration, security, and CDS).

Unsupported Services:

- Security:
  - Transitive Trust in a cell hierarchy.

  - The Public Key Certificate Management API.
  - The Private Key Storage server.
  - Public key login using the OSF DCE 1.2.2 protocol has been superseded by the public key certificate login protocol. However, the security server can still process login requests from other DCE clients that support the OSF DCE 1.2.2 public key login protocol.
- Directory:
  - Hierarchical Cells.
  - cdsalias create, cdsalias connect.
  - Global Directory Services (GDS) are not provided in this release. However, GDS can exist in the same cell and be used for intercell communications, if it is provided by another vendor or product.
- dcecp:
  - **host configure**–Configures a host into the cell as a client or server.
  - **host unconfigure**–Removes the host from the name and security databases.
  - **host start**–Starts DCE on the specified host.
  - **host stop**–Stops DCE on the specified host.

Unsupported Commands:

- CDS:
  - catraverse
  - cds_dbdump
  - cdsd_diag
  - cdsbrowser
- Configuration:

  The **dce_config** script has been replaced by other configuration commands and a SMIT interface. See the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* for more information.
- Security:

  The **sec_salvage_db**, **rlogin**, **rlogind**, **rsh**, and **rshd** commands supplied by OSF.
- Distributed Time Service:

  The **dtss-graph** command, which converts synch trace to PostScript.

Limitations of Supported Services for Accounts Configured to use Public Key Authentication

There are several limitations for accounts configured to use Public Key authentication. These include:

- Public Key accounts cannot use the Password Strength Server.
- The key management API is for use only by applications using the shared-secret key authentication protocol. Applications using public key accounts must use the user-to-user protocol.
- When using GSSAPI, the DCE administrator must set up an account in the DCE registry database for the initiator and the acceptor. The following restrictions apply to the account for the acceptor:
  - The account for the acceptor must be set up to use a key in a keytab file as the account's password.
  - The account for the acceptor cannot be set up to use the user-to-user protocol.
  - The account for the acceptor cannot be set up to use the public key authentication protocol.

  No restrictions apply to the account for the initiator.

Limitations of Supported Services for LDAP Security Servers

With the exception of storing the security registry in an LDAP directory, LDAP security servers operate similarly to legacy security servers. Therefore, you use most of the same commands to administer LDAP security servers. However there are a few legacy security functions that are not supported by LDAP security servers. A list of these unsupported features and limitations follows:

- When you issue the **sec_admin -s** command on a legacy security server, you can provide the replica's name as it appears on the replica list. LDAP security servers do not support this feature. However, you can continue to provide the cell name, the global name, or the network address of the host. For more information abut the sec_admin command, consult the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.
- Unlike legacy security servers, LDAP security servers do not support container ACLs unless the container object is created by DCE. Container ACLs are meaningful only when DCE creates directories. If DCE attributes are attached to existing LDAP objects, the ACL's of the parent are ignored. For more information about container ACLs on legacy DCE security servers, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.
- LDAP does not support DCE aliases. An alias is an alternate name for a primary name.

- Legacy DCE allows principals, groups, and orgs to be renamed using either **dcecp** or **rgy_edit** or by using the **sec_rgy_pgo_rename** API. This functionality is not supported after security data is migrated to LDAP.

  If the master security server is running DCE 3.2 and detects an LDAP migration server configured in the cell, these commands are rejected.

  If the master security server is not running DCE 3.2 and a command is issued to rename a principal, group, or org, the LDAP replicas indicate that they have performed the function when they have not. This keeps the master security server from continuously trying to perform the update. However, the legacy database and the LDAP database will have inconsistent data.
- If DCE objects are located in multiple subtrees, **dcecp** catalog commands search the default DCE subtree under the realm only. If DCE objects are in different subtrees, use LDAP searches for catalog functions.
- It is strongly recommended that you use only case-insensitive names for DCE realms, principals, groups, and organizations. This is because some LDAP servers (such as SecureWay® LDAP) convert any name to single case when processing the name in a DN or an ACL. Converting names to single case can causes improper results with some LDAP operations. For example, you cannot have one primary name stored as joe_programmer and another one stored as JOE_PROGRAMMER because LDAP reads them both as joe_programmer and treats them as the same primary name.

Part 2. Planning for, Installing, Uninstalling, and Migrating to DCE 3.2 for AIX

Chapter 2. Planning

System Requirements

All contents in the DCE 3.2 for AIX product require the retail release of IBM AIX Version 4.3.3 or the 32–bit kernel mode of IBM AIX Version 5.1. See the **README** for updates to the listing of supported versions of AIX. See the *IBM DCE Version 3.2 for AIX: Release Notes* for instructions on accessing the **README**.

In addition to the base operating system requirements, additional AIX software updates might be required. The following describes the AIX software updates that the following DCE filesets require:

- **dce.client.rte.pthreads** requires:
  - On AIX 4.3.3 systems:
    bos.adt.libpthreads 4.3.3.0
    bos.rte 4.3.3.0
    bos.rte.libc 4.3.3.0
  - On AIX 5.1 systems:
    bos.adt.libpthreads 5.1.0.0
    bos.rte 5.1.0.0
    bos.rte.libc 5.1.0.0
- **dce.client.rte** requires:
  - On AIX 4.3.3 systems:
    xlC.rte 4.0.2.0
    bos.net.tcp.client 4.3.3.0
    bos.adt.lib 4.3.3.0
  - On AIX 5.1 systems:
    xlC.rte 5.0.2.0
    bos.net.tcp.client 5.1.0.0
    bos.adt.lib 5.1.0.0
- **dce.tools.appdev.adt** requires:
  - On AIX 4.3.3 systems:
    bos.adt.syscalls 4.3.3.0
    bos.adt.include 4.3.3.0
  - On AIX 5.1 systems:
    bos.adt.syscalls 5.1.0.0

bos.adt.include 5.1.0.0

Notes:

1. Only the 32–bit kernel mode of the AIX 5.1 operating system is supported by DCE 3.2.
2. After you install any of the updates, you must reboot your system. This reboot must occur before you configure or restart DFS.

Disk Space Requirements

See the **README** file for the disk space requirements of the Licensed Programs. Note that the sizes listed are approximations.

Global and Cell Considerations

The purpose of this section is to assist you in planning for the installation and configuration of DCE. DCE provides System Management Interface Tool (SMIT) and configuration utilities to assist you. "Chapter 3. Installing, Uninstalling, and Migrating to DCE 3.2 for AIX" on page 51 and "Configuring DCE" on page 73 describe the configuration process, including installing executable files, setting up a DCE cell, and configuring servers and clients.

This section discusses the following topics:

- "Planning Questions to Consider"
- "Establishing a Cell Name" on page 27
- "The Cell Namespace" on page 29
- "Planning for Access Control" on page 34
- "DCE Naming Considerations for Internationalization" on page 35

Planning Questions to Consider

Note: The information in this section pertains to legacy DCE or an environment where the DCE Security Registry and LDAP Integration feature is being considered. For extended planning considerations on the DCE Security Registry and LDAP Integration feature, refer to the *IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide*.

You need to consider a number of questions when planning for a distributed system.

Keep in mind the following global considerations as you plan for DCE:

- How much do you think your environment will grow in the next few years? Do you anticipate rapid or relatively slow expansion of your network?

If you think your environment will grow rapidly, consider setting up several cells representing smaller units of your organization. You can manage these smaller units as your network expands. As explained previously, members of each cell share a common purpose, and the cell is a unit of administration and security. If you anticipate slow expansion of your network, you may be able to establish one or more cells based on the organization that exists now. Consider how many administrators you will need to maintain your DCE cell, based on anticipated future growth.

- How much information updating do you require? Do the users in your network mainly look up information, or do they create and change information at their workstations?

  If information changes frequently and users in your network depend on the accuracy of that information, you need to consider how much you rely on replication. It is better to go to a central source of information for data that changes frequently. If users look up information, but do not need to change the information that is shared with other users, you can rely more on replicated data.

- Is the most important data the most available? Have you made plans to replicate this data?

  CDS and the Security Service maintain master copies of their respective databases. Each CDS directory can be replicated separately. The Security Service supports replication of the entire registry database. Because other components depend on the information managed by the Security Service and parts of the CDS namespace, that data needs to be available at all times. For example, the special character string **/.:** (the cell root) is stored in CDS and must always be available.

  Keep in mind that while replicating data helps availability, there is a cost in terms of performance and the amount of administration required.

- If your network has a gateway, are servers located on the same side of the gateway as the clients that rely on those servers?

  CDS servers broadcast messages at regular intervals to advertise their existence to CDS clients in the network. Clients learn about servers by listening for these advertisements. Placing servers and the clients that rely on them on the same side of the gateway facilitates efficient updates of information and a quick response to client requests. Additional administration is required if you rely on servers that are not available through the advertisement protocol, which is effective only in a local area network.

  On a LAN that has no CDS servers, proxy advertisers will broadcast the addresses of CDS servers. This means that clients do not need to know the address of a CDS server at the time of configuration. The proxy advertiser will broadcast the address of the CDS server that it was configured with. Additional CDS server addresses can be added using either the **cdscp define server** command or the **dcecp cdscache create** command.

Consider how fast and how expensive links are if you are administering a cell that includes users in different geographic locations. You might want to keep more information locally to reduce your dependence on transmitting information across links.

- Is communication limited to your own cell, or do you need to communicate with other cells?

  For your cell to communicate with other cells, you must:
  - Establish a unique DNS global name for your cell.
  - Define your cell in DNS.
  - Have at least one GDA in your cell or have performed a **cdscp define server** or a **dcecp cdscache create**.

  Note: Global Directory Service (GDS) is not provided with this release of DCE 3.2 for AIX. However, this release can use GDS if it is provided by another product to locate other cells.

  You can set up a special account in your cell's security registry for a foreign cell, indicating that your cell trusts the Authentication Service of the other cell. A special account in the foreign cell's security registry can represent your cell. (For information about setting up these special accounts, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide*.) Even if you do not need to communicate with other cells now, consider whether you will need to communicate with other cells in the future. Be sure to establish a cell name with these future requirements in mind.

Your answers to these questions determine the basic requirements of your user environment. Use these requirements to help you decide on the optimum use of the DCE functions described in this section and the following sections.

- **Resolving Differences between DCE and AIX Standard Accounts**:

  It is strongly recommended that any users and groups defined in the individual system **/etc/passwd** and **/etc/group** files be synchronized with users and groups in the DCE registry. Synchronization can be facilitated with the **passwd_export** and **passwd_import** utilities after initial cell configuration. Any users who are not synchronized between the cell registry and the local files might not realize full benefit of the integration feature. On the other hand, this flexible integration scheme supports wandering users (users who are defined in the DCE registry, but not a local system). If a machine is configured to allow it, those wandering users may log on to the system and obtain DCE credentials and local access based on UNIX-relevant information in the registry.

  When DCE creates the security registry database, DCE includes some standard UNIX principals, groups, and accounts. These do not match those that are included on a typical AIX system. This mismatch can lead to problems if you plan to use the **passwd_export** command to keep **/etc/passwd** and **/etc/group** synchronized with the DCE registry.

If you will include only AIX machines in your cell, you can delete the standard principals, groups, and accounts from the registry and add those that match AIX principals, groups, and accounts.

If your cell will include more types of machines than AIX machines, you can either convert the standard accounts as described in the preceding paragraph or keep the accounts that DCE creates. Then, you can use the **/opt/dcelocal/etc/passwd_override** and **/opt/dcelocal/etc/group_override** files on individual machines to set up standard accounts and groups that match those expected by that machine's operating system. For more information about the override files, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

If you plan to convert standard UNIX accounts in the registry as described here, you should do so immediately after initial cell configuration to reduce the likelihood of producing orphans (objects owned by UUIDs that have been deleted).

- **DCE Application Core Files**: Because DCE applications are multithreaded, their core files become large. Each thread has its own stack and other associated information that are saved in the core file. If you want usable core files from your DCE application while you are developing and testing the applications, make sure you have permission to write large core files. You can use the **ulimit** command to temporarily change the maximum core file size for the current shell process, or you can use the **chuser** command to permanently change the maximum core file size for a particular user.

Establishing a Cell Name

Before you can configure your DCE cell, you need to establish a cell name. This section describes DCE naming syntax, naming conventions, and the procedure for obtaining a cell name.

Global Names

All DCE objects, including applications, machines, and users, have a global name. A global name is meaningful and usable from anywhere in the DCE environment. In DCE, global names begin with the special character string **/...**, which indicates the global root directory.

DNS Global Names: DCE also supports global directory operations through the use of DNS. Following is an example of a global name that uses the DNS format:

/.../seattle.xyz.com/sec/principal/smith

In DNS format, **/.../seattle.xyz.com** is the cell name, followed by a cell namespace entry.

Cell-Relative Names

In the two previous examples, **sec/principal/smith** is that part of the global name that resides in the local cell. The **sec/principal/smith** part of the global name can be used to construct a cell-relative name. Cell-relative names, also known as local names, are meaningful only from within the cell where the name entry exists. Cell-relative names begin with the special character string /.:, which replaces the global part of the name (the cell name). If you are in the **seattle.xyz.com** cell, the following cell-relative name translates to the same global name shown in the previous examples:

/.:/sec/principal/smith

When you are entering a CDS name from the cell where that object is registered, you can use the cell-relative name. However, if you are entering a CDS name from another cell, you must use the global name, beginning with the character string **/...** (the global root).

CDS and DNS naming conventions are described in more detail in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

Choosing a DCE Cell Name

Choosing an appropriate DCE cell name is important for the following reasons:

- All DCE cells that will ever participate in the global namespace must have unique names to differentiate them from cells in other organizations.
- A uniquely identified cell name is critical to the operation of DCE security; this name is the basis for authentication in your cell.
- DNS expects global cell names to have a certain format. Choose a name that conforms to DNS naming conventions.
- DCE does not support cells registered simultaneously in GDS and DNS.

Note that cell names are case insensitive; that is, the name **MyCell** is equivalent to the name **MYCELL**. (When comparing cell names, DCE routines change the names to all lowercase before making the comparison.)

Cell names must not contain an at sign (@). Two cells on the same LAN should not have the same name. Two cells with the same name would be perceived as a single cell. Depending upon configuration, this could cause serious disruptions in the proper functioning of the commonly named cells. Cell names must also be restricted to characters in the DCE Portable Character Set described in *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

The maximum cellname that can be used when configuring a DCE cell will be calculated based on the maximum filename size of 255. The largest filename

that is created by DCE is **<cellname>#<dce_hostname>_ch.checkpoint<10 digit number>.**Due to this limitation, the cell name size will be calculated as **(229 - <length dce_hostname>)>.** This limit will be enforced when the Security Master server or an Initial or Additional CDS Server is configured. The maximum cellname size of 255 will be enforced when configuring a client or any other server into an existing cell.

Obtaining a DCE Cell Name

If you plan to create a private cell and do not ever intend for it to communicate with cells outside your organization, you are not required to obtain a globally unique cell name. However, in order for your cell to communicate with other cells outside your organization, you need to have intercell set up and, before you configure your cell, you need to obtain a globally unique cell name from the GDS or DNS global naming authorities. The name can be one that already exists and is in use, or you can specify that you need a new name. This registration must be completed before you begin to configure the cell namespace. It is recommended that you obtain a unique global name for your cell even if you do not initially use a global directory service to communicate with other cells, so that you can do so in the future.

Defining a Cell in DNS

You can use the **cdscp** subcommand **show cell** to obtain data that you need to create or modify a cell entry in DNS. The data you obtain from the command is what CDS uses to contact servers in foreign cells. Use the **mkreg.dce** command to register cell information with the DNS. For information on setting up the intercell environment, managing intercell naming, and administering a multicell environment, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

The Cell Namespace

An integral part of planning for a DCE cell is understanding the organization of your cell namespace. Consider the following as you plan the organization of a cell in your network:

- Are security requirements maintained?
- Does the organization of the cell facilitate network traffic where data sharing needs are the greatest?
- How will you manage the administrative accounts created for each DCE service during the configuration process?

Determining Cell Boundaries

In DCE, the boundaries of a cell are equivalent to the boundaries of the cell namespace. A small organization can consist of one cell. A large organization can have many cells. The primary factors in determining a cell's boundaries are the common purpose and trust shared by the cell's principals. Principals

within a cell can belong to groups that share the same privileges. Members of a group share the same level of trust and are authorized to perform certain actions.

Because there is a set of administrative tasks associated with setting up and maintaining each cell, it is reasonable to keep the number of cells in your organization to a minimum. However, the level of trust shared by groups of principals is a more important consideration than administrative overhead.

Keeping Cells Stable

Once you decide how many cells you need and where the boundaries of those cells will be, make an effort to keep your cell structure stable. Servers are not easily moved from one cell to another; so, be sure to plan your namespace structure carefully in order to minimize reconfiguration. If you do need to move a host from one cell to another, you must:

- Move server processes from the host.
- Unconfigure the host from the old cell, using the **unconfig.dce** command.
- Use the **config.dce** command to reconfigure the host in the new cell.

Types of Cell Namespace Entries

This section describes the different types of entries that comprise the cell namespace. These entries are created when you follow the default configuration path described in Configuring DCE. The cell namespace can be divided into the following parts:

- The CDS part of the namespace
- The security part of the namespace
- The DFS part of the namespace (the filespace)
- The **dced** (per host) part of the namespace

Each DCE service maintains its own namespace within the DCE cell namespace. DFS maintains its own namespace to ensure consistency among many files. The Security Service maintains its own namespace to ensure that the DCE cell remains secure. Clients of this service query CDS for binding information that enables them to find the security server. The points where the binding information is stored serve as mount points in the CDS namespace for the namespaces that DFS and the Security Service manages. This transition point between two namespaces is called a junction. The **/.:/sec** directory is the junction from the CDS part to the security part of the cell namespace, and the **/.:/fs** directory is the junction from the CDS part to the DFS part of the cell namespace.

The junction **/.:/hosts/***hostname***/config** is the junction from CDS to the **dced** (per host) part of the namespace.

Figure 1 on page 31 shows the top level of the cell namespace. In some cases, the names in the cell namespace are fixed (or well known) and cannot be

changed. In other cases, you can choose a different name from the one listed. In Figure 1, **/.:** and **cell-profile** are well-known names.

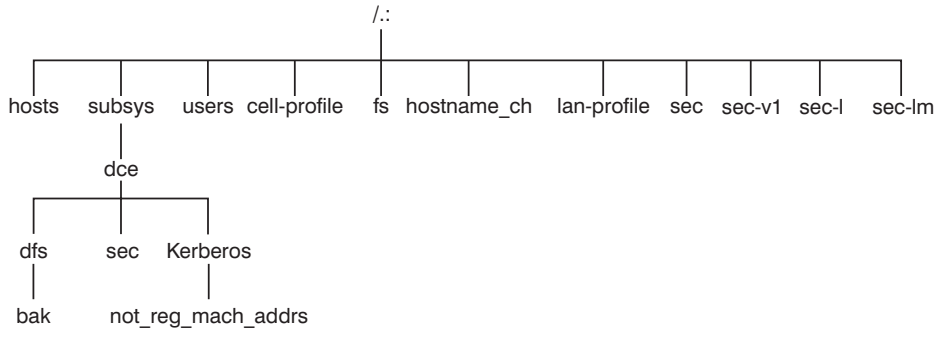

Figure 1. Top Level of the Cell Namespace

You can use the **dcecp**, **rpccp**, **cdscp**, or **cdsli** commands to view the CDS namespace, including the **sec** and **fs** junctions. You can use commands such as **ls** to see the contents of the DFS part of the cell namespace and **dcecp** to see the contents of the security portion of the CDS namespace.

CDS Namespace Entries: The DCE Cell Directory Service is a distributed, replicated database service that is used to store names and attributes of resources located in a DCE cell. This database consists of a hierarchical set of names called the namespace. Each CDS server maintains a portion of the namespace in a local database called a *clearinghouse*, which is optimized for local access. A clearinghouse is designed for relatively few **write** operations (such as creating or deleting directories and objects or exporting binding information), but many **read** operations (such as importing binding information). Note that a clearinghouse is automatically created during the configuration process for a CDS server. See "Configuring the Initial CDS Server" on page 83 for more information.

A CDS database that is distributed and replicated among multiple CDS servers and multiple clearinghouses must be kept consistent. The large number of **write** operations used to replicate and maintain consistency can cause stress-induced CDS failures. Using a large number of replicated CDS directories can also result in stress because updates must be propagated to all the read-only replicas. Use the **cdscp show server** command to display the number of **read** and **write** operations handled by a server since the service was started. This command allows you to monitor the level of activity and adjust the configuration if necessary.

As a directory service, CDS is designed to manage information that does not change often. For example, binding information stored in CDS does not

include endpoints since endpoints change frequently. As you design applications, avoid the need to store highly dynamic data in the CDS namespace.

The CDS namespace contains entries for servers, hosts, CDS clearinghouses (collections of directory replicas stored at a particular server), RPC profiles, RPC groups, and subsystems. The entries have a CDS type of *directory* or *object*, indicating the kind of CDS object to which the name refers. A CDS directory is a container in which objects are stored. CDS uses directories to organize groups of object entries.

Profiles catalogued in the CDS namespace specify a search path through the Directory Service. The cell profile (**/.:/cell-profile**) stores the location of the servers that are available in the cell, regardless of physical location. In a geographically dispersed cell, servers can be located in different cities or even different countries. The LAN profile defines alternate servers that can be used in situations where geographic proximity is important. For example, **/.:/lan-profile** is the default LAN profile used by DTS. This profile contains entries for the DTS server local set. If a cell spans more than one LAN, a profile can be created for each LAN that the cell spans. For example, in a cell that encompasses two LANs, you can direct hosts on one LAN to **/.:/lanA-profile** and hosts on the other LAN to **/.:/lanB-profile**. For information on setting up multiple LAN profiles, see "Configuring DCE" on page 73 .

Security Namespace Entries: The types of security entries are as follows:

principal
: This type of entry contains an individual principal.

principal directory
: This type of entry contains individual principals or one or more principal directories, or both.

group This type of entry contains an individual group.

group directory
: This type of entry contains individual groups or one or more group directories, or both.

org This type of entry contains an individual organization.

org directory
: This type of entry contains individual organizations or one or more organization directories, or both.

policy This type of entry contains a security policy.

When you (or an application) are accessing an entry in the security part of the namespace, the name of the entry alone provides enough information for the

Security Service to work with. For example, the security server knows that the login name is a principal name, registered in the security part of the namespace; **/.:/***principal_name*, **/.../***cell_name/principal_name*, and *principal_name* are all valid ways of representing the name you use to log in.

When you use the **dcecp** command, you specify the type of object you will operate on. For example, to change account information associated with the principal **smith**, you specify that you want to operate on an account. You then enter the principal name **smith**. The **dcecp** command deals with the following types of objects related to security:

- Principals
- Groups
- Organizations
- Accounts
- Xattrschemas

The *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components* explains how to use the **dcecp** command to display information related to principals, groups, organizations, accounts, and xattrschemas.

The **dcecp** also supports operations performed by **acl_edit**. The **acl** object of **dcecp** is used for this purpose. The **dcecp** command requires the object's fully qualified path name when modifying ACLs, as shown in the following example:

```
/.:/sec/principal/smith
```

and not simply the following:

```
smith
```

The following parts of the namespace comprise the security namespace:

/.:/sec/principal
/.:/sec/group
/.:/sec/org
/.:/sec/policy
/.:/sec/xattrschema

CDS Namespace Replication Considerations

Directory replication is the most reliable way to back up the information in your CDS namespace. Because the CDS data is replicated by directory, when you replicate a directory, all of the object's entries in it are automatically replicated. Use the **dcecp** control program to create replicas of directories at a CDS clearinghouse. If you create a clearinghouse in addition to those that are

automatically created on a CDS Server at configuration time, that clearinghouse must be created in the root directory (**/.:**) of the cell namespace.

Follow these guidelines for replicating parts of the cell namespace:

- The root directory (**/.:**) is automatically replicated (without child directories) when you create a clearinghouse.
- You should have at least two replicas of each CDS directory to ensure the entire namespace is available at all times. For further information about backing up CDS information, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

Planning for Access Control

In planning for access control, it is important to keep the level of access control in your cell restrictive enough to ensure that security is maintained. A special set of individuals or a special group can be given permission to create accounts and groups in the root directory of the security namespace. The **acct-admin** group is created when you configure DCE. **acct-admin** is the only group that can create accounts and groups in the root directory of the security space.

While maintaining an adequate level of security in your cell, you also need to consider the requirements of administrators who are maintaining DCE services when you set access control levels.

Following are some of the groups created when you configure DCE using SMIT, **smitty**, or the **config.dce** command:

sec-admin
: This group administers security servers, cell registry functions, and other security functions.

audit-admin
: This group administers Audit servers and related audit functions.

cds-admin
: This group administers CDS servers, CDS replication, and other CDS functions.

dced-admin
: This group administers DCE host servers and ACLs.

dts-admin
: This group administers DTS servers and related DTS functions.

dfs-admin
: This group administers DFS File Servers and related DFS functions.

In addition to the administrative groups, individual users need permission to control some information kept in the registry database. For example, a user needs to be able to change its password, home directory, or login shell.

DCE Naming Considerations for Internationalization

Standard (OSF) DCE, restricts entries in the security namespace, such as principal names, to the characters in the DCE Portable Character Set. See the Architectural Overview of DCE in the *IBM DCE Version 3.2 for AIX and Solaris: Introduction to DCE* for the definition of the DCE Portable Character Set. IBM DCE provides an override capability which enables the use of non-portable characters.

This capability should be used only in environments that are homogeneous with respect to code set and in which all DCE installations support this extension. Security namespace entries that use non-portable characters are guaranteed to work correctly only when the code set of the entire enterprise is the same as that of the process under which the names were created. To enable the use of non-portable security names, the environment variable DCE_USE_NONPORTABLE_NAMES must be set to 1 before DCE is started, in all client and server processes in which DCE security will run.

Certain other names, such as CDS directory names, can also be composed of characters from outside of the DCE Portable Character Set. Because DCE does not perform code set conversion on names, non-portable characters should be used only in environments which are, and will remain, homogeneous with respect to the code set. In environments which are not homogeneous with respect to code set, all DCE names must be restricted to the DCE Portable Code Set.

Subject to the previously mentioned restrictions and to the additional naming rules documented in the *IBM DCE Version 3.2 for AIX and Solaris: Introduction to DCE* and the *IBM DCE Version 3.2 for AIX and Solaris: Application Development Guide—Core Components*, the following names can contain characters outside of the Portable Character Set:

- CDS Object
- CDS Directory
- CDS Attribute
- CDS Link
- RPC idl_byte data
- RPC full name
- Principal
- Group
- Organization
- ERA

- DFS Filename

Client and Server Considerations

This section describes configurations for DCE client machines, the different types of DCE server machines and DCE Application Development Environment machines. A DCE client machine can run client code of every DCE service. DCE server machines are configured to run a certain set of software. This software is made up of at least one daemon and, in some cases, one or more additional programs that comprise the server side of a DCE component. DCE server machines also run the software that makes up the DCE client configuration.

The following topics are provided:
- "Determining Requirements for DCE Client Machines"
- "Determining Requirements for DCE Server Machines" on page 40
- "Determining Requirements for Legacy DCE Migration to an LDAP Security Server" on page 44
- "DCE Administration Utilities" on page 44.

Determining Requirements for DCE Client Machines

This section describes the planning considerations involved in setting up DCE client machines. All DCE machines, including DCE server machines, are also DCE clients.

The following subsections describe the executables that run on a DCE client machine.

RPC Client Programs

A DCE client contains the following RPC programs:

- The **dced** daemon must run on any machine that has an RPC server process that exports an interface with dynamic bindings. The **dced** daemon is used to register binding information.

  The **dced** daemon must be running before you configure any other DCE services that register their endpoints. DCE services need to register their endpoints with **dced**. Only one **dced** daemon can run on a machine at a time, because **dced** uses a well-known port.

  Network interfaces, routing services, and other network services must be available before RPC starts. The **dced** daemon is started by the **start.dce** command. The **start.dce** command can be invoked from **/etc/inittab** by specifying the **-autostart yes** option on the **config.dce** command or by adding **/opt/dcelocal/etc/rc.dce** to this file. This will allow DCE services to be brought up each time the machine boots. See "Using SMIT to Start DCE Immediately and at System Restart" on page 120 for information on the SMIT menu for starting DCE 3.2 for AIX at reboot.

- The DCE control program (**dcecp**) is a utility that allows you to browse, update, add, and delete the RPC attributes of entries stored in the CDS namespace and the endpoints that are managed by local and remote **dced** daemons.

Security Service Client Programs

The **dced** daemon maintains the local machine's principal identity by periodically refreshing the ticket-granting ticket for the machine's principal. This assures that the local root user or any daemon who inherits the machine identity has valid DCE credentials. The **dced** daemon also exports and implements a variety of interfaces, including password and group override support, certification of the security server, and pre-authentication support.

For more information about ticket-granting tickets, see *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

CDS Client Programs

The DCE client runs the following CDS processes:

- The CDS advertiser, the **cdsadv** process, allows applications to access and communicate with **cdsd**. It starts any needed CDS clerks (**cdsclerk**) and creates the cache shared by the local CDS clerks. It receives and caches server broadcasts.
- The **cdsclerk** is an interface between CDS client applications and CDS servers. A clerk must exist on every machine that runs a CDS client application. One **cdsclerk** process runs for each AIX principal on a machine that accesses CDS. The CDS clerk handles requests from client applications to a server and caches the results returned by the server. Because results of the server request are cached, the clerk does not have to go repeatedly to the server for the same information. All CDS clerks on a machine share one cache. One clerk can serve many client applications.
- The DCE control program (**dcecp**) can be used to browse, update and delete CDS entries, and manage the namespace. For more information, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.
- The CDS control program, **cdscp**, is a command interface used to control CDS servers and clerks and manage the namespace and its contents. The **cdscp** command interface was available with previous versions of DCE and is provided to ease migration to the use of the **dcecp** utility. For more information about the CDS control program, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

DTS Client Programs

The DCE client runs the following DTS processes:

- The **dtsd** daemon is set as a client or a server. On a client machine, **dtsd** synchronizes the local clock.

- The **dtscp** program allows you to administer DTS, including configuring the **dtsd** daemon as either a client or a server.

Slim Client Programs

In general, client systems have less available memory than server systems. If a client does not offer DCE services to other systems in the cell, it might not need all of the functions provided by the daemons started by the configuration of DCE software on the client system. The slim client offers the capability of being such a ″reduced″ client in the cell.

Since no information about the slim client is kept in the cell, administrator intervention, that is **cell_admin**, is not required to configure it. Instead use the **config.dce** command to configure the slim client. Use the **start.dce** and **stop.dce** commands respectively to start and stop the slim client. To unconfigure the slim client, use the **unconfig.dce** command.

The slim client option reduces DCE memory consumption on client systems by running a single instance of the CDS clerk with no other DCE daemons. Running a single instance of the CDS clerk is done by starting the clerk with the **-n** option. This starts a clerk without the CDS advertiser. If there are so many other DCE services and functions that can be run, how can a single CDS clerk be sufficient? The answer is that most DCE clients need only the following DCE functions:

- RPC calls (both authenticated and unauthenticated)
- DCE login
- CDS name lookups

For RPC calls and most logins, no DCE daemons are needed. These functions simply use RPC runtime routines and security runtime routines.

For CDS name lookups, only a CDS clerk is necessary. With full DCE, CDS clerks are started by the CDS advertiser, requiring a CDS advertiser to be present. However, in DCE 3.2 for AIX, the **-n** option on the **cdsclerk** command starts a single instance of the CDS clerk without needing the advertiser. This clerk will not terminate after being idle for 20 minutes, as it does in full DCE. Additionally, when the clerk is started in this fashion, it takes over the role of the CDS advertiser in managing the CDS client cache. The clerk will not receive and cache broadcasting CDS servers, a role of the advertiser.

Without an advertiser, the **cdsclerk** can not be managed by **dcecp** or **cdscp**. The following commands will fail:

```
cdscp show clerk
cdscp disable clerk
cdscp show cached clearinghouse
cdscp define cached server
cdscp show cached server
```

```
cdscp clear cached server
dcecp -c cdscache create
dcecp -c cdscache delete
dcecp -c cdscache show -server
dcecp -c cdscache show -clearinghouse
```

The following **dced** services do not run on a DCE slim client:

- **dced Endpoint Mapper Service** must run on any system providing a service that can be accessed through Remote Procedure Calls (RPCs). Such a server is called an RPC server. When a system issues an RPC to an RPC service, it uses the RPC runtime routines to send the request to a specific machine address and asks for the desired RPC service by name. After the RPC reaches the machine where the service resides, the Endpoint Mapper Service maps the RPC service name to the endpoint, or port number, of the specific program providing the service. After the endpoint is known, the client is bound to the specific RPC service, and RPCs can be issued directly to that service.

  Although every DCE client system issues RPCs, most do not need the Endpoint Mapper Service, because they are probably not RPC servers. Therefore, the RPC-related limitation of not running **dced** on a client system is that it cannot be an RPC server.
- **Security Validation Service** provides the functions listed below. If a client system does not need these functions, it does not need the **dced** Security Validation Service. Note that a **dce_login** and authenticated RPCs can still be issued on a system that does not have this service running.
  - **Security Server Certification**.
  - **Third-party pre-authentication during dce_login**.
  - **Keeping the machine context up to date**.
  - **Password and group overrides**.
- **Preferred Security Replica** is not supported for the slim client.
- **System Management Services**:

  The system management functions provided by DCE are listed below. Without **dced**, a client system cannot be remotely managed by means of these functions.
  - **Host Data Management** maintains local files of host data (that includes the host name, cell name, and cell aliases) and a post-processor file. The post-processor file contains program names that are matched to other host data items. **dced** runs the program if the corresponding host data item changes.
  - **Server Control** maintains data that describes the startup configuration and execution state for each server. It can also start or stop particular servers, and enable or disable specific services of servers. This service is not needed on a client that is not running any RPC servers.

– **Key Table Management** allows for the remote maintenance of a server's key tables. This service is not needed on a client that is not running any RPC servers.

Security Integration, **dceunixd**, can run on a slim client. Be aware that because the certification service is not available, when a user logs in, the user's identity cannot be certified to have been issued by a legitimate security server and that security integration on a slim client cannot use password and group overrides. Because the machine context is not available, security integration on a slim client uses unauthenticated access to the registry. In addition, local root users will run unauthenticated because there is no machine context for them to inherit.

Determining Requirements for DCE Server Machines

This section provides information about requirements for the different types of DCE server machines.

Files Installed on DCE Server Machines

The following subsections discuss the files that must be installed on each of the different DCE server machines and the approximate space required. Note that because all DCE servers are also DCE clients, the files described in "Determining Requirements for DCE Client Machines" on page 36 must also be installed on server machines. Therefore, add the appropriate server space requirements to the DCE client machine space requirements to reach the approximate total space requirement for the configuration you are planning.

Security Server Processes and Legacy DCE

Note: The following discussion is based on legacy DCE and can be extended to apply to the DCE Security Registry and LDAP Integration feature.

Every cell has one master DCE Security Service machine and can also have replica DCE Security Service machines. The following processes run on a DCE Security Service master or replica server machine:

- The security server, or **secd** process, implements the Authentication service, the Privilege service, and the Registry service.
- The **sec_create_db** program initializes the security database. The **config.dce** command passes a parameter indicating whether to create a master or replica security server on the machine.
- The DCE control program (**dcecp**) is used for the registry, management, and maintenance of the security server. Optionally, you can use the **sec_admin** program. See "DCE Administration Utilities" on page 44 for descriptions of these programs.

Keep the following considerations in mind when you are planning for security servers:

- The node that runs the master security server must be highly available and physically secure. Consider placing the master security server machine in a locked room and keeping a log to record who accesses the machine.
- Be sure to move the master security server before removing the node from the network or shutting down the node for an extended period of time. Modifications are made to the master security server and propagated to replicas throughout your cell. If the master security server is unavailable, no updates can be made. For more information see "Handling Network Reconfigurations" in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.
- A cell can have only one master security server. If you plan to make one cell out of several existing cells with independent master security servers, you must first merge their registries.
- Keep the following considerations in mind when you are planning for Identity Mapping servers:
  - The DCE security server uses the Identity Mapping server when a DCE user logs in using the public key certificate login feature. You must configure at least one Identity Mapping server to use DCE public key certificate login.
  - Because of the interaction between the Identity Mapping server and the DCE security server, nodes which run Identity Mapping servers should be highly available and physically secure. It is recommended that an Identity Mapping server be run on each node which runs a DCE security server.

For further information about planning for the DCE Security service, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

Since the security registry is replicated in full across all security replicas, if the host that contains the master security server goes down, hosts that have replica DCE Security Servers can still provide registry information; so, consider having a number of replicas in your network. Use factors such as the number of machines in your cell, the reliability of the machines that run security servers, and your cell's available resources to determine how many replica security servers you need to have.

Security Server Processes and LDAP Security Registry

All of the considerations outlined in "Security Server Processes and Legacy DCE" on page 40 need to be considered in an environment where legacy DCE is being or has been migrated to use the LDAP directory to store user data.

For further information about planning for the DCE Security Server using the LDAP directory for user data, see the *IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide*.

CDS and GDA Server Processes

A CDS server stores and maintains object names within a cell and handles requests to create, modify, and look up data. A GDA server enables the cell in which it is running to communicate with other cells.

The following processes run on a CDS server machine:

- The CDS daemon, **cdsd**, is the CDS server process.
- The **cdsadv**, in addition to receiving server broadcasts to find out what servers are available to the local client machine, also sends server advertisements.
- The DCE control program (**dcecp**) for the management and maintenance of the CDS software. In addition, the **cdscp** program for controlling and displaying information about CDS clerks and servers. See "DCE Administration Utilities" on page 44 for descriptions of these programs.

In preparing for CDS, you need to select server nodes that store and maintain the clearinghouses (CDS databases) in the cell.

Keep the following guidelines in mind in order to achieve reliability, optimum performance, and data availability:

- Choose dependable nodes. A CDS server needs minimal downtime and needs to restart quickly. The CDS server needs to be one of the first systems available on the network because client applications and other DCE servers rely on the CDS server for up-to-date information. The CDS server initializes the CDS namespace when you configure DCE.
- Use reliable network connections. This helps to ensure that all servers maintaining directory replicas can be reached when CDS performs a skulk. Skulks are periodic updates that check for consistency across all replicas.
- Consider the size of your cell and how geographically dispersed the cell is when deciding how many CDS servers you need. You should have at least two copies (one master and one read-only replica) of each CDS directory to ensure access to data if one of the servers becomes unavailable.
- Each CDS server maintains at least one clearinghouse. All clearinghouses contain a copy of the root in addition to other directories replicated there.
- You need to make replication decisions based on where the contents of directories are referenced. Put replicas where the contents are read and put masters where the contents are written.

The **gdad** daemon is the GDA server, which sends lookup requests for cell names to the DNS and returns the results to the CDS clerk in the cell that initiated the request.

The GDA can be on the same machine as a CDS server, or it can exist independently on another machine. You can have two or more **gdad** daemons running in a cell to ensure GDA availability.

DTS Server Programs

The DCE client configuration already contains all the files necessary for a DTS server machine, with the exception of the optional time provider.

- The **dtsd** daemon (which can be installed on a DCE client machine) is configured to run as a server. As a server process, **dtsd** synchronizes with other DTS servers, in addition to synchronizing the local clock, as it does on a client machine.
- The **dts**_*device_name*_**provider** specifies the communications between the DTS server process and the time-provider process. For *device_name*, substitute the device you are using, which can be a radio, clock, or modem, or another source of UTC time for DTS. A time provider is optional. If you use a time provider, it must connect to a server process.

Consider the following guidelines when planning your DTS implementation:

- Each cell should have at least three DTS servers. At least three DTS servers are needed in order to detect if one of them is faulty when they are queried for the time. It is preferable to have four or more DTS servers to provide redundancy. The additional servers increase the accuracy of time synchronization. However, increasing the number of servers queried for the time also increases the activity on the network. The administrator must balance the level of accuracy with the amount of network activity.
- A time provider is optional in DTS; however, cells that must be closely synchronized with a time standard need to have at least one time provider.
- Servers need to be located at the sites with the greatest number of different network connections.
- If there are less than three time servers configured in the cell, use one of the following commands:

  ```
  dtscp set servers required n
  (where n is the number of time servers in the cell)
  dcecp -c dts modify -minservers n
  (where n is the number of time servers in the cell)
  ```

  This will prevent a warning message from being logged every time the server attempts to sync.

There are many network configuration decisions that affect DTS planning. The *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components* contains details about the total DTS planning process, including configuration planning for Local Area Networks (LANs), extended LANs, and

Wide Area Networks (WANs) as well as an explanation of the criteria you need to use when selecting a time source for your network to use.

Determining Requirements for Legacy DCE Migration to an LDAP Security Server

For information about planning for the DCE Security Server to use the LDAP directory for registry data, see the *IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide*.

DCE Administration Utilities

This section describes the system administration utilities that can assist you in performing DCE administrative tasks.

DCE Control Program

The DCE control program **dcecp** creates, maintains, and manages RPC, CDS, security, DTS, EMS, and DCED objects. For more information on **dcecp**, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components* and the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.

RPC Administration Programs

The DCE Remote Procedure Call Service provides the following administration utilities:

- The **dced** daemon is used to register binding information.
- The DCE control program (**dcecp**) allows you to browse, update, add, and delete the RPC attributes of entries stored in the CDS namespace and the endpoints that are managed by local and remote **dced** daemons.

See the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components* and the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* for more detailed information about these programs.

DCE Security Service Administration Programs

The DCE Security Service provides the following administration utilities:

- The **dcecp acl** command displays, adds, modifies, and deletes ACL entries for a specific object. The *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* contains detailed information about using the **dcecp acl** command.
- The **dcecp account**, **group**, **organization**, **principal**, **registry**, **user**, and **xattrschema** commands allow you to edit the registry database or the local registry. Almost all editing of the registry database must be done with these commands. The *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* explains the use of the commands.
- The **passwd_import** command allows you to create registry entries based on the group and password files from machines that do not implement DCE Security.

- The **passwd_export** command allows you to update the UNIX **/etc/passwd** and **/etc/group** files with current user information obtained from the registry.
- The **passwd_override** and **group_override** files allow you to establish overrides to the information contained in the registry.
- The **rmxcred** command purges expired tickets from the credentials directory.
- The **dcecp registry** command helps you manage server replicas of the registry, change the master server site, migrate servers to LDAP , and reinitialize a subordinate server. This command also helps you manage the security server and its database. You can perform tasks such as generating a new master key for the database and stopping the security server.

  Note: For information on this topic related to the DCE Security Registry and LDAP Integration feature, refer to the *IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide*.

CDS Administration Programs

CDS provides the following administration utilities:

- The **cdscp** program is described in "CDS Client Programs" on page 37.
- The **cdsli** gives a DCE user the ability to recursively list the namespace of cells.
- The **cdsdel** deletes recursively the namespace of cells. The DCE control program, **dcecp**, can be used to browse, update, and delete CDS entries, and to manage the namespace. It can be used in place of **cdscp**.
- The **mkreg.dce** command enters information about your DCE cell into the database maintained by your domain name server (the **named** daemon).
- The **rmreg.dce** command removes information from the database maintained by your domain name server (the **named** daemon) that were added by the **mkreg.dce** command.

SVC Administration Programs

The **svcdumplog** program prints the contents of a serviceability binary log file as readable text. For more information on **svcdumplog**, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*. The **dce_err** program displays the text associated with a DCE message ID.

DTS Administration Programs

- The **dtscp** command controls the interface you can use to configure and manage DTS. It is already included in the DCE client software.
- The **dtscp** program allows you to administer DTS, including configuring the **dtsd** daemon as either a client or a server.

DCE Web Secure

The DCE Web Secure product extends your Netscape FastTrack or Netscape Enterprise Web server or your iPlanet FastTrack or iPlanet Enterprise Web server to provide DCE authentication to your Web transactions, enabling you to use a Web browser to run Common Gateway Interface (CGI) programs that require DCE credentials, such as DCE Web Administration.

With DCE Web Secure you can provide DCE credentials to a CGI program. For example, with a tcl program called changeusers.tcl that makes batch changes to a group of DCE accounts, administrators would place the changeusers.tcl program into a directory that has been configured for CGI access in the Web server configuration files. When a user runs the CGI program through the Web browser, it gains the DCE credentials under the DCE userid that the Web browser user is logged in as.

DCE Web Administration

With the DCE Web Administration CGI programs, administrators can gain all the functionality of managing DCE users, groups, and organizations, as well as manage permissions.

Application Development Environment

You can configure a DCE machine for the development of DCE applications. This configuration requires adding to the basic DCE client configuration several include (**.h**) and interface specification (**.idl**) files, along with the **idl** compiler. The files and the compiler are included in the **dce.tools.appdev.adt** package available in the DCE for Application Developers licensed program product. You can also use the **sams** utility to include support for messaging and serviceability in your applications. The **sams** utility is included in the **dce.tools.appdev.adt** package.

Location of Installed DCE Files

The files used by DCE are grouped in the following locations:

- The **/opt/dcelocal** subdirectories
- Conventional UNIX subdirectories

Some information needs to be kept locally on a machine for reliability and to ensure security is maintained. For example, when you configure DCE, the file that contains the name of your cell must be on the machine that is being configured. This file is stored in the **/opt/dcelocal** subtree.

The **/opt/dcelocal** subtree is created when you install DCE components.

In some cases, files are installed into directories such as **/usr/lib**, **/usr/bin**, or **/bin** for performance reasons. In other cases, symbolic links can be used from the conventional UNIX subdirectories to **/opt/dcelocal**.

This section contains the following topics:

- "The /opt/dcelocal Subtree"
- "Conventional UNIX Directories"
- "File Locations"
- "File Systems to Create and Mount" on page 49

The /opt/dcelocal Subtree

In order to initially boot a server and configure the cell, the appropriate files for mandatory servers (CDS and security) need to be available on that server machine (in the **/opt/dcelocal** subtree).

Note: It is strongly recommended that copies of the minimum set of programs and data files installed during the default DCE installation procedure be kept locally on server machines for stand-alone operation and emergency maintenance.

The contents of the **/opt/dcelocal** subtree can vary from machine to machine inside a DCE cell to accommodate and serve specific configurations. In addition, every machine must have local access to certain files so each machine can run as a stand-alone system if the machine is disconnected or partitioned from the cell. The appropriate files on DCE servers that have to be local to the server machine must be stored under **/opt/dcelocal**. Client-related data files are stored below **/opt/dcelocal/etc** (static configuration data) and **/opt/dcelocal/var/adm**. All server-specific data files are located in the **/opt/dcelocal/var/***dce-component-name* directory.

The **/opt/dcelocal** subtree is populated and initialized during DCE installation and configuration.

Conventional UNIX Directories

Some files and directories used by DCE are accessible in conventional UNIX directories. These DCE files and directories need to be accessible in conventional locations so users can conveniently access frequently used utilities and data, such as **idl** from the **/usr/bin** directory and **localtime** from the **/etc/zoneinfo** directory. Header files are accessible in **/usr/include** or in its subdirectory, **/usr/include/dce**, and libraries, such as **libdce.a**, are kept in **/usr/lib**.

File Locations

The installation process for DCE 3.2 for AIX places files in the following locations:

/usr/lpp/dce
All DCE files except those in the remainder of this list.

/usr/lpp/dcedoc
All DCE for AIX documentation files and their related tools.

/etc/dce
The following files:
- **rc.dce**
- **dce.clean**
- **rpc.clean**

/etc/dce/rspfiles
Configuration response files.

/etc/zoneinfo
Timezone rules for DTS.

/tmp/dce
Temporary location for configuration processing.

/usr/lib/nls/msg/<*lang*>
Message catalogs, where <lang> is one of the following locales: **en_US**, **es_ES**, **Es_ES**, **ES_ES**, **ja_JP**, **Ja_JP**, **JA_JP**, **ko_KR**, **KO_KR**, **pt_BR**, **PT_BR**, **zh_TW**, **Zh_TW**, or **ZH_TW**.

/usr/include
Include files (mostly under **/usr/include/dce**).

/usr/lib
libdce.a, **libcfgdce.a**, **libdcelibc_r.a**, **libdcepthreads.a**, **libidlcxx.a**, **libcfgdfs.stubs.a**, **libdceweb.a**, **libdceweb4.a**, **libxdsxom.a**, and **libsnmp.a**.

/usr/lib/security
The **DCE** load module for AIX/DCE integrated security operations.

The following symbolic links are established:

subdirectory	**is a symbolic link to**
/opt/dcelocal/	**/usr/lpp/dce**
/opt/dcelocal/var	**/var/dce**
/opt/dcelocal/etc	**/etc/dce**
/opt/dcelocal/tmp	**/tmp/dce**

A link for each of the DCE commands is placed in **/usr/bin**.

In addition, SMIT objects are loaded into the Object Data Manager (ODM) database.

File Systems to Create and Mount

You will probably want to create new AIX JFS file systems in order to use DCE effectively:

/var/dce

All DCE components store information in the **/var/dce** directory. If the **/var** file system fills up, DCE and other subsystems that depend on **/var** (such as the mail and spooler sybsystems) cannot operate correctly.

You should create a new file system mounted over **/var/dce** before you install DCE. You should reserve about 30 megabytes for **/var/dce** for your initial DCE configuration.

/var/dce/directory

The CDS server stores the clearinghouse files, which contain this server's portion of the namespace, and local data in this directory.

If this machine is configured as a CDS server, it is recommended that you create a new file system mounted over **/var/dce/directory** before you install DCE.

You should reserve about 30 megabytes for the server's use.

If you do not plan to create a separate files system for the CDS server, you should add the additional 30 megabytes to **/var/dce**.

/var/dce/security

This is where the security server stores the registry, credentials, and local data. If this machine will be a security server, you should add an additional 10 megabytes to **/var/dce** for the server's use.

Files stored in **/var/dce** are any files particular to the individual machine. You should monitor the space usage in **/var/dce** (and any associated separate files systems) to make sure it does not fill up. To clean up expired credentials files in **/var/dce**, use the **/usr/lpp/dce/bin/rmxcred** command. The DCE Auditing and Servicability facilities also use space in **/var/dce**. See the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* for more information on **rmxcred** and DCE Auditing. See the *IBM DCE Version 3.2 for AIX and Solaris: Problem Determination Guide* for more information on DCE serviceability logs.

Chapter 3. Installing, Uninstalling, and Migrating to DCE 3.2 for AIX

Use the following sections for installation:
- "Installable Packages"
- "Installing DCE 3.2" on page 58
- "Migrating an AIX DCE Cell to DCE 3.2 for AIX" on page 61

Installable Packages

Following is a summary of the installable filesets for the DCE 3.2 for AIX packages. For more detailed descriptions, see "Chapter 1. Overview of DCE 3.2 for AIX" on page 3.

- dce.bundles — DCE Bundles

  dce.bundles.Bnd
  DCE Bundles

- dce.cds — Cell Directory Server Package

  dce.cds.rte
  Cell Directory Services

  dce.cds.smit
  SMIT Cell Directory Services

- dce.client — Base Services Package

  dce.client.rte
  Client Services

  dce.client.core.rte
  Client Services-FOR UPGRADES

  dce.client.core.rte.rpc
  Client RPC Tools-FOR UPGRADES

  dce.client.rte.admin
  Client Administrative Tools

  dce.client.rte.cds
  Client CDS Tools

  dce.client.rte.config
  Client Configuration Tools

  dce.client.rte.pthreads
  Threads Compatibility Library

dce.client.rte.rpc
Client RPC Tools

dce.client.rte.security
Client Security Tools

dce.client.rte.time
Client Time Tools

dce.client.rte.web
DCE Web Secure

dce.client.rte.zones
Client Time Zones

dce.client.smit
SMIT Client Tools

- dce.compat—DCE upgrade package

dce.compat.cds.smit
DCE SMIT Cell Directory Services-FOR UPGRADES

dce.compat.client.core.smit
DCE SMIT Client Tools-FOR UPGRADES

dce.compat.sysmgmt.ems.smit
DCE SMIT Event Management Services-FOR UPGRADES

dce.compat.sysmgmt.snmpagt.smit
DCE SMIT SNMP Subagent-FOR UPGRADES

dce.compat.security.smit
DCE SMIT Security services-FOR UPGRADES

dce.compat.web.admin.smit
DCE SMIT Web Secure Admin-FOR UPGRADES

- dce.doc — DCE Online Documentation

Note: Substitute the appropriate locale name for <*lang*>. DCE 3.2 for AIX supports the following locales: **en_US**, **ja_JP**, **Ja_JP**, **JA_JP**, **ko_KR**, **KO_KR**, **pt_BR**, **PT_BR**, **zh_TW**, **Zh_TW**, or **ZH_TW**.

dce.doc.rte.ascii
DCE ASCII Browser and **dceman**

dce.doc.<*lang*>**.ascii**
DCE ASCII Browser Files

dce.doc.<*lang*>**.html**
HTML Documentation Files

dce.doc.<*lang*>**.pdf**
PDF Documentation Files

- dce.msg.en_US — Messages Package

  Note: Substitute the appropriate locale name for <*lang*>. DCE 3.2 for AIX supports the following locales: **en_US**, **es_ES**, **Es_ES**, **ES_ES**, **ja_JP**, **Ja_JP**, **JA_JP**, **ko_KR**, **KO_KR**, **pt_BR**, **PT_BR**, **zh_TW**, **Zh_TW**, or **ZH_TW**.

  dce.msg.<*lang*>**.client.rte**
  Base Client Messages

  dce.msg.<*lang*>**.cds.smit**
  SMIT Cell Directory Server Messages

  dce.msg.<*lang*>**.client.smit**
  SMIT Base Messages

  dce.msg.<*lang*>**.security.smit**
  SMIT Security Server Messages

  dce.msg.<*lang*>**.sysmgmt.ems.smit**
  DCE SMIT Event Management Messages

  dce.msg.<*lang*>**.sysmgmt.snmpagt.smit**
  DCE SMIT SNMP Subagent Messages

  dce.msg.<*lang*>**.sysmgmt.ems.rte**
  DCE Event Management Services Messages

  dce.msg.<*lang*>**.sysmgmt.snmpagt.rte**
  DCE SNMP SubAgent Messages

  dce.msg.<*lang*>**.sysmgmt.webadmin.rte**
  DCE Web Administration Messages

- dce.priv — Privacy Level Protection Feature Package

  dce.priv.rte
  Privacy Level Protection Feature

  dce.pthreads.rte
  Threads Compatibility Library-FOR UPGRADES

- dce.security — Security Server Package

  dce.security.rte
  Security Services

  dce.security.smit
  SMIT Security Services

- dce.sysmgmt — DCE System Management Package

  dce.sysmgmt.ems.rte
  DCE Event Management Services

dce.sysmgmt.ems.smit
DCE SMIT Event Management Services

dce.sysmgmt.snmpagt.rte
DCE SNMP Subagent

dce.sysmgmt.snmpagt.smit
DCE SMIT SNMP Subagent

dce.sysmgmt.webadmin.rte
DCE Web Secure Admin GUI

- dce.tools — DCE Tools for Application Developers

  dce.tools.admin.rte
  Administration Tools

  dce.tools.appdev.adt
  Application Development Tools

- dce.web—DCE Web Secure upgrades

  dce.web.admin.rte
  DCE Web Secure GUI for upgrades

  dce.web.secure.rte
  DCE Web Secure for upgrades

- dce.xdsxom — X.500 API Library Package

  dce.xdsxom.rte
  X.500 API Library

Requisite Software

Table 1 lists the DCE 3.2 for AIX filesets in the order in which they are installed. See the *IBM DCE Version 3.2 for AIX: Release Notes* for the latest requisite levels of software.

Note: Those software names beginning with "dce" are at the same release level as the shipped DCE product.

Table 1. Installation filesets and requisite software

Fileset You Are Installing	Prerequisite[1], Corequisite[2], and Instreq[3] Software Names	Prerequisite, Corequisite, or Instreq Software Description
dce.bundles — Bundles Package		
dce.bundles	N/A	N/A
dce.client — Base Services Package		
dce.client.rte.pthreads	bos.rte[1]	AIX Base Operating System (BOS) Runtime

Table 1. Installation filesets and requisite software (continued)

Fileset You Are Installing	Prerequisite[1], Corequisite[2], and Instreq[3] Software Names	Prerequisite, Corequisite, or Instreq Software Description
dce.client.rte.pthreads	bos.rte.libpthreads[1]	libpthreads Library
dce.client.rte.pthreads	bos.rte.libc[1]	libc Library
dce.client.rte	bos.net.tcp.client[1]	TCP/IP Client Support
dce.client.rte	xlC.rte[1]	C ++ Set for AIX Application Runtime
dce.client.rte	bos.adt.lib[1]	Base Application Development Libraries
dce.client.rte	dce.client.rte.pthreads[1]	DCE Threads Compatibility Library for AIX
dce.client.rte	dce.client.rte.config[2]	DCE Client Configuration Tools
dce.client.rte	dce.client.rte.security[2]	DCE Client Security Tools
dce.client.rte	dce.client.rte.cds[2]	DCE Client CDS Tools
dce.client.rte	dce.client.rte.time[2]	DCE Client Time Tools
dce.client.rte	dce.client.rte.zones[2]	DCE Client Time Zones
dce.client.rte	dce.client.rte.admin[2]	DCE Client Administrative Tools
dce.client.rte	dce.client.rte.rpc[2]	DCE Client RPC Tools
dce.client.rte.admin	dce.client.rte[1]	DCE Client Services
dce.client.rte.cds	dce.client.rte[1]	DCE Client Services
dce.client.rte.config	dce.client.rte[1]	DCE Client Services
dce.client.rte.rpc	dce.client.rte[1]	DCE Client Services
dce.client.rte.security	dce.client.rte[1]	DCE Client Services
dce.client.rte.time	dce.client.rte[1]	DCE Client Services
dce.client.rte.zones	dce.client.rte[1]	DCE Client Services
dce.client.smit	dce.client.rte[1]	DCE Client Service
dce.client.rte.web	dce.client.rte[1]	DCE Client Services
dce.priv — Privacy Level Protection Package		
dce.priv.rte	dce.client.rte[1]	DCE Client Services
dce.security — Security Server Package		
dce.security.rte	dce.client.rte[1]	DCE Client Services

Table 1. Installation filesets and requisite software (continued)

Fileset You Are Installing	Prerequisite[1], Corequisite[2], and Instreq[3] Software Names	Prerequisite, Corequisite, or Instreq Software Description
dce.security.smit	dce.security.rte[1]	DCE Security Server
dce.security.smit	dce.client.smit[1]	DCE SMIT Client Tools
dce.cds — Cell Directory Server Package		
dce.cds.rte	dce.client.rte[1]	DCE Client Services
dce.cds.smit	dce.cds.rte[1]	DCE Cell Directory Server
dce.cds.smit	dce.client.smit[1]	DCE SMIT Client Tools
dce.doc — DCE Online Documentation		
dce.doc.rte.ascii	dce.client.rte.pthreads[1]	DCE Threads Compatibility Library for AIX
dce.doc.en_US.ascii	dce.doc.rte.ascii[1]	DCE ASCII Browser and **dceman**
dce.doc.en_US.html	N/A	Web Browser
dce.doc.en_US.pdf	N/A	PDF Files
dce.tools — DCE Tools for Application Developers		
dce.tools.admin.rte	dce.client.rte[1]	DCE Client Services
dce.tools.appdev.adt	dce.client.rte[1]	DCE Client Services
	bos.adt.syscalls[1]	System Calls Application Development Toolkit
	bos.adt.include[1]	Base Application Development Include Files
dce.xdsxom — X.500 API Library Package		
dce.xdsxom.rte	dce.client.rte.pthreads[1]	DCE Threads Compatibility Library for AIX
dce.sysmgmt — DCE System Management Package		
dce.sysmgmt.ems.smit	dce.sysmgmt.ems.rte[1]	DCE Event Management Services
dce.sysmgmt.ems.smit	dce.client.smit[1]	DCE SMIT Client Tools
dce.sysmgmt.ems.rte	dce.client.rte[1]	DCE Client Services
dce.sysmgmt.snmpagt.rte	dce.client.rte[1]	DCE Client Services

Table 1. Installation filesets and requisite software (continued)

Fileset You Are Installing	Prerequisite[1], Corequisite[2], and Instreq[3] Software Names	Prerequisite, Corequisite, or Instreq Software Description
dce.sysmgmt.snmpagt.smit	dce.sysmgmt.ems.rte[1]	DCE Event Management Services
dce.sysmgmt.snmpagt.smit	dce.client.smit[1]	DCE SMIT Client Services
dce.sysmgmt.webadmin.rte	dce.client.rte.web[1]	DCE Web Secure
dce.msg.en_US — Messages Package		
dce.msg.en_US.client.rte	dce.client.rte[3]	DCE Client Services
dce.msg.en_US.cds.smit	dce.cds.smit[3]	DCE SMIT Cell Directory Server
dce.msg.en_US.client.smit	dce.client.smit[3]	DCE SMIT Client Tools
dce.msg.en_US.security.smit	dce.security.smit[3]	DCE SMIT Security Server
dce.msg.en_US.sysmgmt.ems.smit	dce.sysmgmt.ems.smit[3]	DCE SMIT Event Management
dce.msg.en_US.sysmgmt.snmpagt.smit	dce.sysmgmt.snmpagt.smit[3]	DCE SMIT SNMP Subagent
dce.msg.en_US.sysmgmt.ems.rte	dce.sysmgmt.ems.rte[3]	DCE Event Management Services
dce.msg.en_US.sysmgmt.snmpagt.rte	dce.sysmgmt.snmpagt.rte[3]	DCE SNMP Subagent
dce.msg.en_US.sysmgmt.webadmin.rte	dce.sysmgmt.webadmin.rte[3]	DCE Web Administration

Table 1. Installation filesets and requisite software (continued)

Fileset You Are Installing	Prerequisite[1], Corequisite[2], and Instreq[3] Software Names	Prerequisite, Corequisite, or Instreq Software Description
Notes: For the Messages Package, the following language filesets can be substituted for the **en_US** fileset: **es_ES**, **Es_ES**, **ES_ES**, **ja_JP**, **Ja_JP**, **JA_JP**, **ko_KR**, **KO_KR**, **pt_BR**, **PT_BR**, **zh_TW**, **Zh_TW**, or **ZH_TW**. For the DCE Online Documentation, the following language filesets can be substituted for the **en_US** fileset: **ja_JP**, **Ja_JP**, **JA_JP**, **ko_KR**, **KO_KR**, **pt_BR**, **PT_BR**, **zh_TW**, **Zh_TW**, or **ZH_TW**. [1]Prerequisite filesets must be installed prior to the fileset that you want to install. (The fileset can not be installed before the prerequisite fileset.) [2]Corequisite (Coreq) filesets must be available to be installed when the fileset that you want to install is installed. (The order in which the filesets are installed is not important.) You can not deinstall one fileset with deinstalling others that it coreqs, or that coreq it. [3]Filesets that instreq other filesets will only be installed if the instreq'ed fileset is installed or available for installation. The fileset that is instreq'ed has no dependency upon the fileset that instreq's it. (The order in which the filesets are installed is not important.) The fileset that instreq'ed another can be removed. The fileset that was instreq'ed can not be removed without removing the one that instreq'ed it.		

Note: To use and configure LDAP with the security servers, you must install the **ldap.client.rte 3.2.1.0** fileset or higher.

Installing DCE 3.2

This section discusses DCE 3.2 installation.

Software Processes to Stop

If you are upgrading an existing installation, manually shut down any applications that run on DCE, and run **dce.clean all** to stop any running DCE processes.

Running the Easy Installation Program

With the AIX operating system, you can install software more easily by using software bundles. A software bundle contains a list of software products that are suited for a particular use. The following procedure shows how to install DCE for AIX using this feature. It can be run either at the system console or remotely and run either under X Windows or from an ASCII terminal.

1. Log in as root.
2. Type:

```
smitty easy_install_bundle
```

3. Specify the installation device or directory for the installation media by pressing **<F4>** to display a list. Select the input device you want.
4. Press **<Enter>**.
5. Select **Media-Defined** and press **<Enter>**.
6. Press **<Enter>** again to accept the Easy Installation selections.
7. At the **Are You Sure** prompt, press **<Enter>** again.The DCE Bundles are installed into **/usr/sys/inst.data/sys_bundles**.
8. Press **<F3>** to cancel.
9. Press **<F4>** to get the new list of software bundles to install.

   App-Dev
   Client
   DCE-CDS
   DCE-Client
   DCE-Management
   DCE-Security
   DCE-Starter
   DCE-Tools
   Media-Defined
   Pers-Prod
   Server

   See Table 2 for the content of each DCE bundle selection.
10. For each bundle that you want to install:
    a. Select the bundle name.
    b. Press **<Enter>**.
    c. Press **<Enter>** again.
    d. Press **<F3>** to cancel.
    e. Press **<F4>** to return to the list of bundles, if you still want to install additional bundles.

Table 2. DCE Software Bundles

Software Bundle	Bundle Name	Name and Filesets
DCE-CDS	DCE-CDS.bnd	**CDS Bundle** dce.cds.smit dce.cds.rte

Table 2. DCE Software Bundles (continued)

Software Bundle	Bundle Name	Name and Filesets
DCE-Client	DCE-Client.bnd	**DCE Client Bundle** dce.client.rte dce.xdsxom.rte dce.client.smit
DCE-Management	DCE-Management.bnd	**DCE Management Bundle** dce.sysmgmt.ems.rte dce.sysmgmt.snmpagt.rte dce.sysmgmt.ems.smit dce.sysmgmt.snmpagt.smit dce.client.rte.web
DCE-Security	DCE-Security.bnd	**DCE Security Bundle** dce.security.rte dce.security.smit dce.client.rte.security
DCE-Starter	DCE-Starter.bnd	**DCE Starter Bundle** dce.client.rte dce.xdsxom.rte dce.client.smit dce.security.rte dce.cds.rte dce.security.smit dce.cds.smit
DCE-Tools	DCE-Tools.bnd	**DCE Tools Bundle** dce.tools.admin.rte dce.tools.appdev.adt

Special Installation Instructions

For information about installation procedures, see the *RISC System/6000 Installation Guide* which you received with your AIX Operating System. This book contains information on the **installp** command.

Migrating an AIX DCE Cell to DCE 3.2 for AIX

Because DCE 3.2 for AIX is dependent upon AIX 4.3.3 or AIX 5.1, one of these versions must be installed on your machines to migrate from DCE 1.3, 2.1, 2.2, or 3.1 (any of these versions with or without PTFs) for AIX to DCE 3.2 for AIX. You can do this migration without a reconfiguration of your existing DCE cell by using the following procedures. You are not required to migrate your machines in a specific order, but please pay close attention to the limitations on DCE Security server functionality as described in "Migrating DCE Security Replicas (Legacy DCE only)". Read this entire section before beginning the migration procedure.

Before Migrating

1. In DCE 3.2 for AIX, each workstation in a DCE cell keeps configuration information about the DCE clients and servers running on the local machine.
2. In case you need to recover your pre-migration DCE configuration, back up all data in the following directories and subdirectories:

   ```
   /opt/dcelocal/var
   /opt/dcelocal/krb5
   /opt/dcelocal/etc
   ```

   or run the **dceback** command found on the DCE CD-ROM. The **dceback** command is documented in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.
3. Install AIX 4.3.3 or AIX 5.1, including the PTFs which are prerequisites for DCE. The prerequisites are listed in the *IBM DCE Version 3.2 for AIX: Release Notes*. Use the **Migration** version of AIX Install.

Migrating DCE Clients

1. Stop DCE. See "Stopping DCE Daemons" on page 121 for information on stopping DCE.
2. Install DCE 3.2 for AIX. Choose the same server and client packages that you had installed for your previous level of DCE for AIX on this machine.
3. Start DCE by running **start.dce**. **start.dce** will invoke **migrate.dce** to migrate all DCE configuration data to the DCE 3.2 for AIX format. Because AIX Security Integration (**dceunixd**) was not supported by pre-DCE 2.2 configuration tools, it will be migrated only if it can be detected in the **/etc/inittab** file. If DFS was previously configured and **start.dce** is specified, **migrate.dfs** will migrate the DFS data.

Migrating DCE Security Replicas (Legacy DCE only)

DCE security replica servers can be migrated using the steps documented in "Migrating DCE Clients" on page 61. We recommend that you migrate all security replicas in your cell prior to enabling DCE 3.2 for AIX function on your master security server. When planning your migration, keep the following limitations in mind:

1. If security replicas are migrated prior to the migration of the master security server, they will run with only your previous level of DCE for AIX function enabled. When the DCE 3.2 for AIX function is enabled on the security master using the **dcecp** command (given in Step 3 on page 63 under "Migrating the DCE Security Master (Legacy DCE only)"), these security replica servers will also enable the DCE 3.2 for AIX function.
2. If the DCE 3.2 for AIX function is enabled on the master security server prior to the migration of all security replicas, any replicas which are running your previous level of DCE will be shut down. These security replicas cannot support DCE 3.2 for AIX function.

Migrating DCE CDS Servers

1. Ensure that all CDS master directory replicas located on this machine are replicated on at least one other CDS server machine in the cell. If you want to support updates to these CDS directories during the migration process, move these master directory replicas to another CDS server.
2. Perform the tasks described in "Migrating DCE Clients" on page 61.

Migrating the DCE Security Master (Legacy DCE only)

1. To minimize the impact to ongoing cell operations, ensure that at least one security server replica is running before you commence. This will support continuing security server **query** operations, though **update** operations will not be supported during the time the master security server is down.

   If the machine which is your master security server is also a CDS server, ensure that all CDS master directory replicas located on this machine are replicated on at least one other CDS server machine in the cell. If you want to support updates to these CDS directories during the migration process, move these master directory replicas to another CDS server.
2. Perform the tasks described in "Migrating DCE Clients" on page 61.

   At this point in the migration process, all your previous level of DCE for AIX functions remain operable, but DCE 3.2 for AIX functions are not yet enabled.

Note: If you intend to enable the Security Registry and LDAP Integration feature, please refer to the *IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide* for information on setting the security registry level.

3. While logged in as the cell administrator, enable DCE 3.2 for AIX function, by performing the following steps:
   a. Check the registry version by issuing the following command:
      ```
      dcecp -c registry show
      ```
   b. If the version shown is not secd.dce.1.2.2, issue the following command:
      ```
      dcecp -c registry modify -version {secd.dce.1.2.2}
      ```
   c. If the cell will be supporting Public Key certificate login, issue the following command:
      ```
      dcecp -c registry modify -version {secd.dce.1.2.2a}
      ```
   d. If enabling LDAP, see the *IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide*. If the DCE 3.2 cell is migrated to exploit the DCE Security Registry and LDAP Integration feature, the Registry version number will change to secd.dce.1.3.

   It is recommended that you perform these steps only after all security replica servers in your cell have been migrated to DCE 3.2 for AIX. For further information, refer to the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

   Note: When the Security registry version is changed, any security replica servers in your cell that can not run at this new Security registry version will be shutdown. OSF DCE Security registry versions secd.1.2.2 and secd.1.2.2a are supported by DCE 2.2 for AIX, DCE 3.1 for AIX and Solaris, DCE 3.2 for AIX and Solaris, and DCE for Windows NT® V2.2. OSF DCE Security registry version secd.1.3 is only supported by DCE 3.2 for AIX and Solaris.

4. After you have migrated the master security server, you need to validate any intercell accounts that exist in the DCE registry. This can be done by using the following command while logged in as the cell administrator:
   ```
   /usr/bin/dcecp -c account modify krbtgt/cell_name -change {acctvalid
   yes}
   ```

   where *cell_name* is the name of the foreign cell. If you do not validate these accounts, intercell access from non-AIX OSF 1.2.2 clients to the foreign cell will fail.

Migrating DTS Servers

To correctly migrate DTS servers, follow the tasks described in "Migrating DCE Clients" on page 61. Note that any time providers in use on the system will not be recognized by the migration tool. To continue using a time provider, it might be necessary to manually reconfigure it after the migration is completed.

Migrating Password Strength Servers

The following information will be useful when migrating a machine with a configured Password Strength server.

The Password Strength server shipped with DCE 3.1 is an enhanced Password Strength server. This new Password Strength server will overwrite the pwd_strengthd in **/opt/dcelocal/bin**. The previous version of pwd_strengthd that was in **/opt/dcelocal/bin** will be saved as **/usr/lpp/save.config/usr/lpp/dce/bin/pwd_strengthd**.

- **DCE-provided Password Strength server**

  If you were using the Password Strength server shipped with a previous version of DCE (pwd_strengthd), unless additional manual migration steps are performed, the new Password Strength server will only support the level of function shipped in the previous release. In order to operate in enhanced mode, the new Password Strength server requires additional configuration steps that were not performed in previous releases of DCE. These additional steps cannot be performed by the migration process because cell administrator authority is required to perform the steps. To enable the new features, do one of the following:

  – Update your existing Password Strength server configuration.

    1. Log in as the cell administrator.
    2. Run

       ```
       /opt/dcelocal/bin/migrate_pwd_strengthd
       ```

  – Reconfigure the Password Strength server.

    On the Password Strength server machine:

    1. Run

       ```
       unconfig.dce pw_strength_svr
       ```

    2. Run

       ```
       config.dce pw_strength_svr
       ```

  See *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components* for information on how to enable the Enhanced Password Strength server rules for DCE users' passwords.

- **Customized Password Strength servers**

If you have written your own Password Strength server using an executable name other than **/opt/dcelocal/bin/pwd_strengthd**, no special migration steps are required.

If you were using a customized version of pwd_strengthd in **/opt/dcelocal/bin**, you can find your customized program in the saved location indicated previously. If you wish to continue using your customized version of pwd_strengthd, it is recommended that you move your customized pwd_strengthd to a new location then reconfigure it. If you do this, installing future versions of DCE will not overwrite your customized pwd_strengthd. Alternately, you can replace the installed pwd_strengthd with the saved pwd_strengthd. Note that the rules that you have defined will not be enforced until you restart DCE after replacing the installed Password Strength server.

If you wish to reconfigure your customized Password Strength server, perform the following steps on the Password Strength server machine:

1. Copy your customized password strength server from **/usr/lpp/save.config/usr/lpp/dce/bin/pwd_strengthd** to a location other than **/opt/dcelocal/bin**.
2. Unconfigure your password strength server using the following command:

   ```
   unconfig.dce -pwdstr_principal pwd_strengthd pw_strength_svr
   ```

3. Reconfigure your password strength server using the following command:

   ```
   config.dce -pwdstr_principal <principal name> \
   -pwdstr_cmd <fully qualified exe> \
   -pwdstr_arg <command line args> \
   pw_strength_svr
   ```

   where <*principal name*> is the principal name that your Password Strength server uses. The default for the previous release was pwd_strengthd.

   Notes:

   a. If you apply more than one command line argument to the **-pwdstr_arg** option, the arguments must be enclosed by double quotation marks (″). For example:

      ```
      -pwdstr_arg "-v -d"
      ```

   b. If you want to specify a password strength principal other than pwd_strengthd, specify both the **-pwdstr_principal** option and the **-pwdstr_arg -server_princ** option. For example:

      ```
      -pwdstr_principal pwd_server
      -pwdstr_arg "-server_princ pwd_server"
      ```

Migrating Identity Mapping Servers

1. To correctly migrate an Identity Mapping Server, follow the first two steps of "Migrating DCE Clients" on page 61.
2. If and only if you replaced **/opt/dcelocal/bin/idmsd** with your own idmsd daemon, copy your backup idmsd from **/usr/lpp/save.config/usr/lpp/dce/bin**.
3. Start DCE by running **start.dce**.

Migrating CDS Cached Servers

When migrating a previous DCE version of an AIX DCE system that has an intercell connection established by use of the **cdscp define cached server** command, the **config.dce** command must be used after installing the new level of DCE and before using **start.dce** in order to preserve knowledge of the cached server. Use the **-cds_replica_list** option to specify the cds servers. This is a quoted list that can have multiple servers separated by spaces. Either the hostname or the IP address can be used in this list. For example:

```
config.dce -cds_replica_list "server1 server2"
```

By running this command, the list of cds servers is preserved outside of the cds cache. This information will be used to update the cds cache each time DCE is restarted. The **-cds_replica_list** option can be used at any time to update the list of cds servers. The **config.dce** command will perform a define cached server for each new cds server. When servers are removed, the cache will not be updated until the cache is cleaned up.

If the previous step isn't performed, the intercell connection can be reinstated by repeating the original **cdscp define cached server** command or the corresponding **dcecp cdscache create** command.

Note: The **-cds_replica_list** option was supported in DCE 3.1. If this CDS server information was saved prior to this migration, it does not have to be saved again. The CDS server information can be updated by re-running the preceding command.

Migrating DCE Web Secure to current version of DCE for AIX

If you had DCE Web Secure configured for a release prior to DCE 3.1, then you must migrate your Web server to the current release before it can be functional. An attempt will be made to migrate the dceweb configuration information when any of the following commands are run:

- **migrate.dceweb**
- **start.dce**

- **stop.dce**
- **config.dce**
- **unconfig.dce**

Also, if the specific dceweb server you are trying to configure or unconfigure has not been successfully migrated, an attempt will be made to migrate the dceweb configuration information when any of the following commands are run:

- **mkdceweb**
- **rmdceweb**

For a specific previously-configured DCE Web Secure/Admin Netscape server to be functional, the following conditions must be met:

- The Netscape server in question must be running.
- DCE must be running.
- The DCE WebSecure/Admin configuration data for the server must be migrated.
- The Netscape server in question must have been stopped and restarted after the configuration data was migrated.

The **migrate.dceweb** process will stop and restart the DCE Web Secure/Admin Netscape server that it was able to successfully migrate.

It is recommended that you run **migrate.dceweb** and resolve any problems that are encountered before attempting to start DCE.

Migrating DCE Web Secure/Admin in IBM DCE V2.1 for AIX to IBM DCE V3.2 for AIX

There is a situation where a DCE Web Secure/Admin server can be configured and working, but all the information pertaining to the server was not stored in the **/opt/dcelocal/web/etc/servers** file. Because of this, some previously-configured DCE WebSecure/Admin servers might not be migrated up to the IBM DCE V3.2 for AIX level.

Check the **/opt/dcelocal/web/etc/servers** file to see if all of your configured servers are listed. An entry in the servers file should look like:

```
my_server admin Enterprise 3.62 /netscape/suitespot
```

The first field is the server name. The second name is the DCE WebSecure/Admin type (secure for secure only, admin for admin/secure). The third field is the Netscape server type (Enterprise or FastTrack). The fourth field is the Netscape version number. The last field is the server's Netscape home directory.

If you are looking at this file before migration has been performed, some of these lines might be split. **Do not attempt to fix these split lines.** The servers file should only be modified by DCE programs (unless otherwise directed by DCE documentation).

If any of your previously configured DCE WebSecure/Admin servers are not listed, do the following:

1. For each missing dceweb server, run the following as root:

   ```
   add_dceweb_entry <servername> <dceweb_type> <ns_home>
   ```

   For example:

   ```
   add_dceweb_entry my_server admin /netscape/suitespot
   ```

2. After all entries have been successfully added to the servers file, run the DCE Web Secure/Admin migration program, **migrate.dceweb**.

The **add_dceweb_entry** program will do the following:

- If the server entry already exists in the **/opt/dcelocal/web/etc/servers** file, the entry will be displayed.
- If the server entry was successfully added to the servers file, the new entry will be displayed.
- If an error occurred, the program will display the data that is causing the problem along with three question marks (???).

The **add_dceweb_entry** program is only intended to add entries to the servers file. It will not update entries that are already there.

Duplicate and Obsolete Filesets Listed on System Management Interface Tool (SMIT) Installation Menus

Several DCE filesets were renamed in the IBM DCE V3.1 for AIX release. When an upgrade installation is done, filesets are installed based on what is currently installed. Since the filesets were renamed, filesets with the new names are not found on the system. To allow upgrades, "dummy" filesets with the old names were created. These filesets corequisite the filesets with the new names. They do not install any files.

When installing through SMIT using the install latest or install all menus, you will see two of the same fileset listed. One of these is the real fileset with a new fileset name, the other is the "dummy" fileset. The "dummy" fileset will have **FOR UPGRADES** in the fileset description. For example, **dce.client.core.rte** has been renamed to **dce.client.rte**. There is a "dummy" fileset named **dce.client.core.rte**. The description that you will see for the dce.client package will be:

```
dce client
+ 3.2.0.0 DCE Client Administrative Tools
+ 3.2.0.0 DCE Client CDS Tools
+ 3.2.0.0 DCE Client Configuration Tools
+ 3.2.0.0 DCE Client RPC Tools
+ 3.2.0.0 DCE Client RPC Tools - FOR UPGRADES
+ 3.2.0.0 DCE Client Security Tools
+ 3.2.0.0 DCE Client Services
+ 3.2.0.0 DCE Client Services - FOR UPGRADES
+ 3.2.0.0 DCE Client Time Tools
+ 3.2.0.0 DCE Client Time Zones
+ 3.2.0.0 DCE SMIT Client Tools
+ 3.2.0.0 DCE Threads Compatibility Library
+ 3.2.0.0 DCE Web Secure
```

Notice that there are two entries for ″DCE Client Services″ and ″DCE Client RPC Tools″. One entry contains ″- FOR UPGRADES″. This is the ″dummy″ fileset.

The ″dummy″ fileset options do not have to be selected. They are there for use by the update_all menu.

Once DCE is installed, the ″dummy″ filesets do not need to be on the system. Most of them will be removed by other filesets. Sometimes, a few might be left on the system. The order in which they are installed might cause some of them to be left. They can be left on the system or removed.

See “Appendix C. Dummy Filesets” on page 133 for a list of ″dummy″ filesets.

Migrating an Existing DCE Cell to Exploit DCE Security Registry and LDAP Integration

DCE Security Registry and LDAP integration is a new feature in IBM DCE 3.2 for AIX. This feature enhances IBM DCE by removing security information stored in the DCE specific database known as the security registry and storing this information in an LDAP directory.

See the *IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide* for information on migrating an existing DCE cell to exploit this new feature.

Uninstalling DCE 3.2

Before uninstalling DCE for AIX, Version 3.2, you must unconfigure your machine. See "Unconfiguring DCE Components" on page 112 for information about unconfiguration.

At the SMIT **Software Installation and Maintenance** panel:

1. Select **Software Maintenance and Utilities**.
2. Select **Remove Software Products**.
3. Select **The Software to Remove**.

   Type the name of the software you want to uninstall. Press **<F4>** to display a list of all the installed software.
4. Select **OK** or **Do**.

Suggested Reading

For information about AIX installation procedures, see the *RISC System/6000 Installation Guide*, which you received with your AIX Operating System.

For information on configuring a DCE cell, see the "Configuring DCE" on page 73 and the **config.dce** command in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.

For information about unconfiguring individual DCE components, see the **unconfig.dce** command in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.

For information on configuring and unconfiguring DCE Web Secure, see the **mkdceweb** and **rmdceweb** commands in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.

Part 3. Configuring, Starting, and Stopping DCE 3.2 for AIX

Chapter 4. Configuring DCE 3.2 for AIX Servers and Clients

Configuring DCE

The following sections describe creating and configuring a DCE cell:
- "Overview of Configuration"
- "Initial Cell Configuration" on page 80
- "Further Cell Configuration" on page 94
- "Unconfiguring DCE Components" on page 112

These sections include server and client components for the following DCE services: Security Service, Cell Directory Service (CDS), Distributed Time Service (DTS), Remote Procedure Call (RPC), and Global Directory Agent (GDA).

To access the SMIT menus for DCE, select **Communications Applications and Services** from the main SMIT menu. Then, select **DCE (Distributed Computing Environment)** from the Communications Applications and Services submenu.

For information on setting up the intercell environment, managing intercell naming, and administering a multicell environment, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.

Overview of Configuration

The configuration of a DCE cell occurs in two phases. During the first phase, or *initial cell configuration*, certain tasks must be performed to initialize the cell. During the second phase, generic tasks can be performed to configure (or reconfigure) additional features into the cell.

A DCE cell requires the following components:
- One Security server
- One CDS server

It is recommended that there also be at least one DTS server (although three or more DTS servers are preferred for accuracy of time synchronization).

The Security and CDS servers must be configured to initialize any cell. After the cell is up and running, you generally will not have to repeat any of these configuration tasks.

Additional components that can be configured into a cell are the following:

- DCE clients (**dced**, **cdsadv**, **cdsclerk**, **dtsd**)
- Secondary CDS servers (**cdsd**)
- Replica Security Servers (**secd**)
- Audit Services (**auditd**)
- Global Directory Agents (**gdad**)
- DTS Services (**dtsd**)
- Simple Network Management Protocol (**dceagtd**)
- Event Management Service (**emsd**)
- Password Strength Server (**pwd_strengthd**)
- Security Integration (**dceunixd**)
- Identity Mapping Service (**idmsd**)
- Name Service Interface Daemon (**nsid**)
- DCE Web Secure

The configuration of these additional components is a task you can perform at any point throughout the lifetime of the cell after initialization.

Keep the following items in mind when you are configuring a cell:

- For better performance and reliability install the master security server and the initial CDS server on different machines.
- Clients can be configured in one of three ways:

  Split Configuration
  This type of configuration is used when the DCE cell administrator is unlikely to have root user access to every machine in the cell. It is comprised of two distinct sets of operations:

  admin This type of configuration updates the namespace and security registry with information about the new client. The cell administrator must run the **config.dce** command from a machine within the existing cell. It can not be run from the new client machine. The cell administrator does not need root user authority to run the admin portion of configuration.

  local This type of configuration creates the necessary files on the local machine and starts the daemons for the new client. The admin part of **config.dce** must have been run first, or the local configuration will fail when trying to contact the cell. The user must have root authority on the machine, but does not need to have any authority in the DCE cell. The following components do not require that the admin part of **config.dce** be run first: slim client, audit, integrated login, and RPC.

  Full Configuration
  This type of configuration is the default. Full configuration includes

both admin and local configuration steps. The DCE cell administrator must have root authority on the local machine being configured into the cell.

- Before configuring a machine into a cell, make sure that the machine's clock is within five minutes of the cell's master security server's clock. If the machine's clock is skewed more than five minutes, authentication errors might result, and configuration might fail. If you have already configured at least one DTS server in the cell, you can use the **-sync_clocks** flag to perform the synchronization for you automatically.
- If you want to reconfigure a particular component (or an entire machine) with new parameters, you must unconfigure it to remove the existing configuration before setting up the new configuration.
- To enable intercell communication using GDA, you must also register the cell's name into a global directory, such as the Domain Name System (DNS). For information on the intercell environment, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*.
- You can perform initial and additional configuration tasks using System Management Interface Tool (SMIT). SMIT uses interactive menus (rather than a command-line interface) to guide users through configuration and other system management tasks. The following sections provide step-by-step procedures for cell configuration using SMIT.

DCE 3.2 for AIX also provides the following commands to perform these same configuration tasks at the command line:

chpesite
: Updates the **pe_site** file, which contains the addresses of the security servers that you use.

clean_up.dce
: Cleans up recreatable database files, cache files, and credential files. This command is intended to be used if problems are encountered when trying to start DCE.

config.dce
: Configures and starts DCE components. This command provides for a *split configuration of clients*. Administrative configuration and local configuration can be performed separately. See "Further Cell Configuration" on page 94 for more information.

kerberos.dce
: Creates the host principals, FTP principals, and key table entries used to support the Secure Remote commands.

migrate.dce
Migrates DCE configuration data from previous releases for use with the current release. There is no need to reconfigure when installing a new release of DCE.

mkdceweb
Configures DCE Web Secure and DCE Web Administration into a Netscape FastTrack or Netscape Enterprise Web server or an iPlanet FastTrack or iPlanet Enterprise Web server.

mkreg.dce
Adds information about a DCE cell into the DOMAIN namespace.

rmdceweb
Unconfigures DCE Web Secure and DCE Web Administration from a Netscape FastTrack or Netscape Enterprise Web server or an iPlanet FastTrack or iPlanet Enterprise Web server.

rmreg.dce
Removes information about a DCE cell from the DOMAIN namespace (DNS).

show.cfg
Displays the local host's DCE or DFS configuration. The **dce** and **dfs** options allow display of only DCE or DFS information.

start.dce
Starts the configured DCE components. This command makes sure that all components are started in the correct order.

stop.dce
Stops the configured DCE components. This command makes sure that all components are stopped in the correct order.

unconfig.dce
Removes configurations of DCE components. This command provides for a *split unconfiguration*, with which administrative configuration and local configuration can be performed separately. See "Further Cell Configuration" on page 94 for more information.

Note: In IBM DCE V2.2 for AIX and above, the **rmdce** command is mapped to **unconfig.dce**. There are several changes in behavior from IBM DCE V2.1 for AIX and below to IBM DCE V2.2 for AIX and above for the **rmdce -o** command. Because of the additional component dependency checking which has been added to **unconfig.dce**, the **-F** flag must be specified when using **rmdce** to perform administrative unconfiguration of a DCE client. The correct command is:

```
rmdce -o admin -h <dce_hostname> -F all_cl
```

Additionally, there are cases where **rmdce** will fail with the following message:

```
0x11315417: Neither the IP host name nor IP address of the machine was sp
0x11315463: The attempt to determine the host IP name and address was uns
                    Reissue the unconfig.dce command specifying the -host_
```

To handle this, the **-i** option has been added to **rmdce.**

-i identifier
Identifies the host to be unconfigured using the **-o** administration option. May be a machine hostname or TCP/IP address.

For detailed information on these commands, refer to the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.

Note: The **lsdce** command now calls the command **show.cfg**. The format of the output from the **lsdce** command has changed. The output now looks like the following:

```
Gathering component state information...

Component Summary for Host: witsend.austin.ibm.com

Component Configuration State
Security Master server  Configured
Security Client  Configured
RPC  Configured
Initial Directory server  Configured
Directory client  Configured

The component summary is complete.
```

The **lsdce** command passes a flag to **show.cfg** to bypass the daemon state checks. The **show.cfg** command runs slower than the old **lsdce** command because it needs to determine the running state of the daemon. Refer to the documentation in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* on **show.cfg** for more information.

User-Supplied Commands

The DCE 3.2 for AIX **config/unconfig/start/stop** code now provides support for user-supplied commands. User-supplied commands can be executed before, after, or before and after configuration, unconfiguration, start and stop of DCE. The intent of this support is to allow you to run your own commands without having to modify the scripts that are shipped with the DCE product. When a future release of this product is installed, your user-supplied commands will automatically run with the new release.

Note: User commands are user-supplied scripts, executables, and so on, that the DCE **config/unconfig/start/stop** commands call. The DCE commands are TCL scripts and can only call commands in a new shell. Because of this, you can not set environment variables in your scripts and expect them to be available to the calling DCE command. If you need to set environment variables, please put these in the **/etc/environment**, or set them before running the DCE configuration commands. If the preceding methods of setting environment variables are not acceptable, the environment variables can be added directly to the **user_cmd.tcl** script. Environment variables set in this manner are available to all of the DCE **config/unconfig/start/stop** scripts. Use the following TCL syntax:

```
set env(YOUR_ENV_VAR) "your_env_var-value"
```

where YOUR_ENV_VAR is the environment variable that you want to set.

Perform the following:

Write your command to do what you need. When executed, the configuration commands (**config.dce**, **unconfig.dce**, **start.dce**, and **stop.dce**) set the environment variable, **"callers_cmd_line"** (including all the parameters with the exception of the cell administrator's password), to the command line. For example, when configuring DCE, if the command executed is:

```
"config.dce -cell_name mycellname -admin_pwd -dce-
sec_srv cds_srv"
```

The **callers_cmd_line** environment variable is set to:

```
"-cell_name mycellname -admin_pwd <****>
sec_srv cds_srv"
```

This environment variable might be useful to your command script.

Create the file **/opt/dcelocal/tcl/user_cmd.tcl**. This file should contain the appropriate subset of the following entries:

For DCE:

- **set pre_config_dce** — the full path to your pre-DCE configuration command and any arguments.
- **set pre_config_dce_fail_on_error** — **$TRUE** or **$FALSE** to indicate whether the **config.dce** should fail (or not) if your command fails.
- **set post_config_dce** — the full path to your post-DCE configuration command and any arguments.

- **set post_config_dce_fail_on_error** — **$TRUE** or **$FALSE** to indicate whether the **config.dce** should fail (or not) if your command fails.
- **set pre_unconfig_dce** — the full path to your pre-DCE unconfiguration command and any arguments.
- **set pre_unconfig_dce_fail_on_error** — **$TRUE** or **$FALSE** to indicate whether the **unconfig.dce** should fail (or not) if your command fails.
- **set post_unconfig_dce** — the full path to your post-DCE unconfiguration command and any arguments.
- **set post_unconfig_dce_fail_on_error** — **$TRUE** or **$FALSE** to indicate whether the **unconfig.dce** should fail (or not) if your command fails.
- **set pre_start_dce** — the full path to your pre-DCE start command and any arguments.
- **set pre_start_dce_fail_on_error** — **$TRUE** or **$FALSE** to indicate whether the **start.dce** should fail (or not) if your command fails.
- **set post_start_dce** — the full path to your post-DCE start command and any arguments.
- **set post_start_dce_fail_on_error** — **$TRUE** or **$FALSE** to indicate whether the **start.dce** should fail (or not) if your command fails.
- **set pre_stop_dce** — the full path to your pre-DCE stop command and any arguments.
- **set pre_stop_dce_fail_on_error** — **$TRUE** or **$FALSE** to indicate whether the **stop.dce** should fail (or not) if your command fails.
- **set post_dce** — the full path to your post-DCE stop command and any arguments.
- **set post_stop_dce_fail_on_error** — **$TRUE** or **$FALSE** to indicate whether the **stop.dce** should fail (or not) if your command fails.

Note: Use # to include a comment on its own line. Use ;# to include a comment on a line of code.

The configuration, unconfiguration, start, and stop codes look for the **/opt/dcelocal/tcl/user_cmd.tcl** file and the following variable names:

pre_config_dce
post_config_dce
pre_unconfig_dce
post_unconfig_dce
pre_start_dce
post_start_dce
pre_stop_dce
post_stop_dce

pre_config_dce_fail_on_error
post_config_dce_fail_on_error
pre_unconfig_dce_fail_on_error
post_unconfig_dce_fail_on_error
pre_start_dce_fail_on_error
post_start_dce_fail_on_error
pre_stop_dce_fail_on_error
post_stop_dce_fail_on_error

Examples:

```
#Make sure that App XYZ is not already configured
#config.dce should fail if /usr/bin/APP_XYZ_config_check fails
set pre_config_dce "/usr/bin/APP_XYZ_config_check"
set pre_config_dce_fail_on_error $TRUE

#The following command runs the App XYZ config command
#App XYZ must be configured after DCE
#config.dce will NOT fail if /usr/bin/APP_XYZ_config fails
set post_config_dce "/usr/bin/APP_XYZ_config -arg1 arg1_value -arg2 arg2_value"

#The following command runs the App XYZ start command
#App XYZ must start after DCE
#start.dce will NOT fail if /usr/bin/APP_XYZ_start fails
set post_start_dce "/usr/bin/APP_XYZ_start"
set post_start_dce_fail_on_error $FALSE

#Stop App ABC before stopping DCE
set pre_stop_dce "/usr/bin/APP_ABC_stop"
#stop.dce will fail if /usr/bin/APP_ABC_stop fails
set pre_stop_dce_fail_on_error $TRUE
```

Environment Variables

Environment variables are variables used by DCE that customers can set themselves. See the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Introduction* for more information about DCE environment variables.

Initial Cell Configuration

To initialize a cell, you must perform these basic tasks in order:

1. Configure the master security server machine. See "Configuring the Master Security Server" on page 81.
2. Configure the initial CDS server machine. See "Configuring the Initial CDS Server" on page 83.
3. Configure a CDS client on the master security server. See "Configuring a CDS Client on the Master Security Server" on page 93.

In the procedures that follow, ensure that the *dce_hostname* of each machine is unique within the cell. The *dce_hostname* is the name that is listed in the hosts directory (**hosts/***dce_hostname*) in the namespace. The **config.dce** command allows you to assign a *dce_hostname* independent of a machine's host name on the network. By default the host name of the machine is used.

Attention: If you attempt to configure two machines that have the same *dce_hostname*, you will have to unconfigure and reconfigure DCE on both machines. If one of these machines is either the Security server or the initial CDS server, you will have to unconfigure and reconfigure DCE on *every* machine in the cell.

The following sections provide detailed procedures for performing these initial configuration tasks using **smitty**. See the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* for information on commands that you can use to perform these same configuration tasks at the command line.

Configuring Servers

This section discusses the following:

- "Configuring the Master Security Server"
- "Configuring the Initial CDS Server" on page 83

Configuring the Master Security Server

To configure the master security server for a cell, perform the following steps on the machine that is designated as the master security server:

Note: The security server can be started with as many as 512 executor threads. By default, the security server is started with 10 executor threads. Increasing the number of executor threads might increase the throughput of the security server. However, if the security server is running on AIX 4.3.3 or AIX 5.1, the maximum number of executor threads should not exceed 200. Refer to "Changing Arguments Passed to DCE Daemons at Restart" on page 120 and **secd** security server information in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.

To configure the master security server using **smitty**:

1. As root, start **smitty** with the **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Servers
   ```

2. Select the **SECURITY Server** option, and press **<Enter>**.
3. Select the **primary** option, and press **<Enter>**.

4. At the **CELL name** prompt, type the name of the cell. For each cell, the first time you run this menu and type a name at this prompt, you establish the cell name. You will use the cell name later in other configuration menus.

   Note: The cell name cannot be changed during the lifetime of the cell.
5. If you do not want to use the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
6. If you do not want to use the default cell administrator's UNIX ID, type the cell administrator's UNIX ID at the **Cell ADMINISTRATOR's account UNIX ID** prompt. The default is 100.
7. Type the *dce_hostname* for this machine in the **Machine's DCE HOSTNAME** field. If no name is selected, the TCP/IP hostname, including domain, will be used.
8. Tab to select **yes** or **no** in the **Start components at System restart** field to indicate that the DCE components should or should not be automatically started at system reboot.
9. Tab to select **yes** or **no** in the **Clean up DCE at System restart** field to indicate that **clean_up.dce** should be run at system reboot. **clean_up.dce** removes all recreatable DCE files (that is, cred files, cdscache files, and so on). This will be done before DCE is restarted.
10. In the **Protocol** field, press **<F4>** to list the protocols. Select the protocols that DCE will be configured within, and press **<Enter>**. Once selected, the same protocols must be used for subsequent configurations.
11. **dced** will automatically update the pe_site file with security server information. Type a time to wait between pe_site updates at the **Minutes to wait between pe_site file updates** field.

    ```
       0          (zero) turn automatic update of the pe_site file off
    10 - 1440    10 to 1440 minutes
    ```

    The default is 1440 minutes (24 hours).
12. Type the security server name for this machine in the **Security Server Name** field. If no name is selected, the DCE hostname will be used.
13. Tab to select **yes** or **no** in the **Use CERTIFICATE based login** field to indicate that PK certificate authentication should be enabled.
14. If PK certificate authentication is to be used, type the fully-qualified path for the file containing the **ENTRUST PROFILE for the security server**.
15. If PK certificate authentication is to be used, type the fully-qualified path for the **ENTRUST INITIALIZATION file**.
16. If PK certificate authentication is to be used, type the **ENTRUST PROFILE password** for the DCE security server.

17. If you want to merge your current **/etc/passwd** and **/etc/group** files into your new DCE registry, you may encounter UNIX ID conflicts. To avoid these conflicts, type new values (if necessary) in the appropriate fields for the starting point and the maximum value for UNIX IDs assigned to principals, groups, and organizations. The defaults are the values displayed. Because UNIX IDs cannot be changed once the security service has created accounts, you should set the values for the starting point and the maximum value for UNIX IDs now, when the registry is first created.
18. Press **<Enter>** to select **Do**.
19. When prompted, type the password to be assigned to the initial account created in the registry database. Make a note of the password for the cell administrator's account, because it is required to perform other configuration tasks. This password is also assigned to the DCE account for root that is granted privileged authority by DFS.

To configure the master security server from a command line, at the command prompt type:

```
config.dce -cell_name <cell_name> [-sec_server_name <security_server>]
[-cell_admin <cell_admin id>] [-admin_pwd <admin_password>]
[-min_princ_id <min_principal_id>] [-min_group_id <min_group_id>]
[-min_org_id <min_org_id>] [-max_unix_id <max_UNIX_id>]
[-no_pesite_update] [-pesite_update_time <update_time>]
[-autostart yes | no] [-clean_autostart yes | no] [-protocol tcp udp]
[-certificate_based_login yes | no ] [-kdc_profile <kdc_profile>]
[-kdc_ini_file <kdc_ini_file> ] [-kdc_passphrase <kdc_passphrase>]
[-group_rsp_path <filename>] [-rsp_file <filename>]
sec_srv
```

At this point, **dced** (RPC and a security client) and the master security server are configured on the machine. You can return to this machine later to configure CDS and DTS.

Configuring the Initial CDS Server

There can be only one *initial* CDS server for each cell. To configure the initial CDS server for a cell, perform the following steps on the machine that is designated as the initial CDS server:

To configure the initial CDS server using **smitty**:

1. As root, start **smitty** with the **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   or perform the following sequence of **smitty** menu options:

```
1. Communications Applications and Services
2. DCE (Distributed Computing Environment)
3. Configure DCE/DFS
4. Configure DCE/DFS Servers
```

2. Select the **CDS (Cell Directory Service) Server** option, and press **<Enter>**.
3. Select the **initial** option, and press **<Enter>**.
4. If this machine is not the master security server, perform the following steps. If this machine is the master security server, these fields are automatically filled in with the proper values.
   a. At the **CELL name** prompt, type the name of the cell.
   b. If you are not using the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
   c. Type the *dce_hostname* for this machine in the **Machine's DCE HOSTNAME** field. If no name is selected, the TCP/IP hostname, including domain, will be used.
   d. Tab to select **yes** or **no** in the **Start components at System restart** field to indicate that the DCE components should or should not be automatically started at system reboot.
   e. Tab to select **yes** or **no** in the **Clean up DCE at System restart** field to indicate that **clean_up.dce** should be run at system reboot. This will be done before DCE is restarted.
   f. In the **Protocol** field, press **<F4>** to list the protocols. Select the protocols that DCE will be configured within, and press **<Enter>**. Once selected, the same protocols must be used for subsequent configurations.
   g. At the **MASTER SECURITY Server** prompt, type the TCP/IP hostname or IP address of the machine that is the master security server. (The TCP/IP hostname or IP address is not necessarily the same as the *dce_hostname*.)
   h. **dced** will automatically update the pe_site file with security server information. Type a time to wait between pe_site updates at the **Minutes to wait between pe_site file updates** field.

      ```
         0          (zero) turn automatic update of the pe_site file off
      10 - 1440    10 to 1440 minutes
      ```

      The default is 1440 minutes (24 hours).
5. Tab to select **yes** or **no** in the **Synchronized Clocks** field to indicate that the local machine's clock should or should not be synchronized with a DTS server.

6. Type the name of the time server you want to use for synchronization at the **Time Server to Synchronize Clocks with** prompt. The time server can be any DTS server.
7. If the cell will contain multiple LANs and require the use of global DTS servers, or if the preferred security replica support will be enabled, type the name of the LAN profile this machine should use at the **LAN PROFILE** field. However, if this is the same machine as the security server, the field will already be filled in.
8. Press **<Enter>** to select **Do**.
9. When prompted, type the cell administrator's password.

To configure the initial CDS server from a command line, at the command prompt type:

```
config.dce [-cell_name <cell_name>] [-cell_admin <cell_admin id>]
[-admin_pwd <admin_password>] [-sec_master <master_security_server>]
[-autostart yes | no] [-clean_autostart yes | no] [-protocol tcp udp]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[-wrap_audit_trail yes | no]
cds_srv
```

At this point, **dced** (RPC and a security client), the initial CDS server, and a CDS clerk are configured on the machine. (If this machine is the master security server, only the initial CDS server and a CDS clerk are actually configured in this section.) You can return to this machine later to configure DTS.

Note that a clearinghouse is automatically created when you configure a CDS server. Although it is possible to define multiple clearinghouses for a CDS server, you should have only one per CDS server during normal operation. If you are moving a clearinghouse from one CDS server to another, however, you can temporarily define a second clearinghouse on the original server. See the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components* for more information on moving a clearinghouse.

Configuring Clients

This section discusses the following:

- "Configuring DCE Clients" on page 86
- "Admin Client Configuration" on page 86
- "Local Client Configuration" on page 87
- "Full Client Configuration" on page 89
- "Slim Client Configuration" on page 91
- "Configuring a CDS Client on the Master Security Server" on page 93

Typically, you need to configure many clients into a DCE cell. Configuring clients entails two distinct sets of operations:

- Tasks that require *cell administrator* authority within the DCE cell
- Tasks that require *root user* authority on the machine that is to be configured as a DCE client.

These tasks are separated into a *split configuration of clients* because a DCE cell administrator is unlikely to have root user access to every machine in a cell.

Configuring DCE Clients

The DCE clients can be configured in one of three ways: full, admin, or local.

Split client configuration for security clients (**sec_cl**) and CDS clients (**cds_cl**) is a two-part process. (The cell administrator might not have root access to the client machines, or the root user might not have cell administrator access.) The two parts are the following:

- The cell administrator runs the **admin** portion from any machine in the cell to update the CDS namespace and security registry.
- The root user of the client machine runs the **local** portion to create necessary files and to start client daemons for all client components.

Admin Client Configuration

To do the **admin** portion of configuring a DCE client, the cell administrator performs the following steps from any machine in the cell:

To do the **admin** portion of configuring a DCE client using **smitty**:

1. Start **smitty** with the **mkdceclient** fastpath:

   ```
   smitty mkdceclient
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Clients
   ```

2. Select **admin only configuration for another machine**, and press **<Enter>**.
3. Press **<F4>** for a list of DCE clients. For **admin** configuration, the only selections are **all_cl**, **sec_cl**, **cds_cl**, **dts_cl**, and **nsid**. All other client configuration is done on the client machine and does not require cell administrator authority. Select the name of the DCE client you want to configure, and press **<Enter>**.
4. If you are not using the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
5. Type the *dce_hostname* for the client machine in the **Machine's DCE HOSTNAME** field. If no name is selected, the TCP/IP hostname, including domain from Step 6 on page 87, will be used.

6. Identify which machine is being configured as a client by typing its TCP/IP hostname or IP address in the **Machine's name or TCP/IP address** field.
7. If the cell contains multiple LANs and requires the use of global DTS servers, type the name of the LAN profile the client machine should use at the **LAN PROFILE** prompt.
8. Press **<Enter>** to select **Do**.

To do the **admin** portion of configuring a DCE client from a command line, at the command prompt type:

```
config.dce -config_type admin -host_id <machine identifier>
[dce_hostname <dce_hostname> [-cell_admin <cell_admin id>
[-admin_pwd <admin_password>] [-lan_profile <profile>]
[-protocol tcp udp] [-group_rsp_path <filename>]
[-rsp_file <filename>]
sec_cl | cds_cl | dts_cl
```

At this point, the namespace entries and security registry database have been updated. It is now necessary to run the **local** portion of configuration to complete the process.

Local Client Configuration

To do the **local** portion of configuring a DCE client (after the **admin** portion is completed, when appropriate), perform the following steps as root on the client machine:

To do the **local** portion of configuring a DCE client using **smitty**.

1. Start **smitty** with the **mkdceclient** fastpath:

   ```
   smitty mkdceclient
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Clients
   ```

2. Select **local only configuration for this machine**, and press **<Enter>**.
3. Press **<F4>** for a list of DCE clients. For **local** configuration, the selections are **all_cl**, **rpc**, **sec_cl**, **cds_cl**, **dts_cl**, **slim_cl**, and **dce_unixd**. Select the name of the DCE client you want to configure, and press **<Enter>**.
4. Ensure that the **CELL name** field is filled in with the appropriate values.
5. Type the *DCE_hostname* for this machine at the **Machine's DCE HOSTNAME** prompt. If no name is selected, the TCP/IP hostname, including the domain, will be used.

6. Tab to **yes** or **no** in the **Start components at System restart** field to indicate that the DCE components should or should not be automatically started at system reboot.
7. Tab to select **yes** or **no** in the **Clean up DCE at System restart** field to indicate that **clean_up.dce** should be run at system reboot. This will be done before DCE is restarted.
8. At the **Protocols** prompt, press **<F4>** to list the protocols. Select the protocols that DCE will be configured within, and press **<Enter>**. Once selected, the same protocols must be used for subsequent configurations. If you will be configuring any DFS components on this machine, you must use the **udp** protocols.
9. If the machine is on a separate LAN from any CDS server and cannot send broadcast packets to a CDS server or if you are not sure if the machine is on a separate LAN, type the TCP/IP hostname or IP address of the master security server at the **MASTER SECURITY Server** prompt.
10. **dced** will automatically update the pe_site file with security server information. Type a time to wait between pe_site updates at the **Minutes to wait between pe_site file updates** field.

    ```
       0          (zero) turn automatic update of the pe_site file off
    10 - 1440    10 to 1440 minutes
    ```

    The default is 1440 minutes (24 hours).
11. If the machine is on a separate LAN from any CDS server and cannot send broadcast packets to a CDS server or if you are not sure if the machine is on a separate LAN, type the TCP/IP hostname or IP address of the CDS server at the **CDS Server (If in a separate network)** prompt.
12. If you want to define other CDS servers in the CDS cache, type the TCP/IP hostnames or IP addresses of these CDS servers at the **List of additional CDS Servers** field.
13. Tab to select **yes** or **no** in the **Synchronized Clocks** field to indicate that the local machine's clock should or should not be synchronized with a DTS server.
14. Type the name of the time server you want to use for synchronization at the **Time Server to Synchronize Clocks with** prompt. The time server can be any DTS server.
15. If the machine is on a separate LAN and you want the CDS client to rebroadcast the location of the CDS server, at the **Rebroadcast CDS Server Location** prompt tab to select **yes**. The default is **no**.
16. Press **<Enter>** to select **Do**.

To do the **local** portion of configuring a DCE client from a command line, at the command prompt type:

```
config.dce -config_type local
[-cell_name <cell_name>] [-dce_hostname <dce_hostname>]
[-sec_master <master_security_server>] [-cds_server <cds_server>]
[-no_pesite_update] [-pesite_update_time <update_time>]
[-time_server <server id>] [-sync_clocks yes | no] [-autostart yes | no]
[-clean_autostart yes | no] [-protocol tcp udp] [-proxy]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[-num_dce_unixd <number>] [cache_lifetime <minutes>]
[-cds_replica_list <list_of_cds_servers>] [-wrap_audit_trail yes | no]
client_components
```

At this point, the selected clients are configured on the machine.

Full Client Configuration

If you are both the *cell administrator* and the *root user* of a machine currently being configured as a client, you can perform a **full client** configuration, which incorporates both the **admin** and **local** portions of configuration.

To perform the **full** configuration of a DCE client, perform the following steps as root on the client machine:

To perform the **full** configuration of a DCE client using **smitty**:

1. Start **smitty** with the **mkdceclient** fastpath:

   ```
   smitty mkdceclient
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Clients
   ```

2. Select **full configuration for this machine**, and press **<Enter>**.
3. Press **<F4>** for a list of DCE clients. For **full** configuration, all clients are available for selection. Select the name of the DCE client you want to configure, and press **<Enter>**.
4. Ensure that the **CELL name** field is filled in with the appropriate values.
5. If you are not using the default *cell_admin*, type the name of the cell
6. Type the *DCE_hostname* for this machine at the **Machine's DCE HOSTNAME** prompt. If no name is selected, the TCP/IP hostname, including the domain, will be used.
7. Tab to select **yes** or **no** in the **Start components at System restart** field to indicate that the DCE components should or should not be automatically started at system reboot.

8. Tab to select **yes** or **no** in the **Clean up DCE at System restart** field to indicate that **clean_up.dce** should be run at system reboot. This will be done before DCE is restarted.
9. At the **Protocols** prompt, press **<F4>** to list the protocols. Select the protocols that DCE will be configured within, and press **<Enter>**. Once selected, the same protocols must be used for subsequent configurations. If you will be configuring any DFS components on this machine, you must use the **udp** protocols.
10. If the machine is on a separate LAN from any CDS server and cannot send broadcast packets to a CDS server or if you are not sure if the machine is on a separate LAN, type the TCP/IP hostname or IP address of the master security server at the **MASTER SECURITY Server** prompt.
11. **dced** will automatically update the pe_site file with security server information. Type a time to wait between pe_site updates at the **Minutes to wait between pe_site file updates** field.

```
   0          (zero) turn automatic update of the pe_site file off
10 - 1440    10 to 1440 minutes
```

    The default is 1440 minutes (24 hours).
12. If the machine is on a separate LAN from any CDS server and cannot send broadcast packets to a CDS server or if you are not sure if the machine is on a separate LAN, type the TCP/IP hostname or IP address of the CDS server at the **CDS Server (If in a separate network)** prompt.
13. If you want to define other CDS servers in the CDS cache, type the TCP/IP hostnames or IP addresses of these CDS servers at the **List of additional CDS Servers** field.
14. Tab to select **yes** or **no** in the **Synchronized Clocks** field to indicate that the local machine's clock should or should not be synchronized with a DTS server.
15. Type the name of the time server you want to use for synchronization at the **Time Server to Synchronize Clocks with** prompt. The time server can be any DTS server.
16. If the cell will contain multiple LANs and require the use of global DTS servers, or if the preferred security replica support will be enabled, type the name of the LAN profile this machine should use at the **LAN PROFILE** field.
17. If the machine is on a separate LAN and you want the CDS client to rebroadcast the location of the CDS server, at the **Rebroadcast CDS Server Location** prompt, tab to select **yes**. The default is **no**.
18. Press **<Enter>** to select **Do**.

To perform the **full** configuring of a DCE client from a command line, at the command prompt type:

```
config.dce -config_type full
[-cell_name <cell_name>] [-dce_hostname <dce_hostname>]
[-admin_pwd <admin_password>] [-cell_admin <cell_admin id>]
[-sec_master <master_security_server>] [-cds_server <cds_server>]
[-lan_profile <profile>] [-pesite_update_time <update_time>]
[-no_pesite_update]  [-time_server <server id>] [-sync_clocks yes | no]
[-autostart yes | no] [-clean_autostart yes | no] [-protocol tcp udp]
[-proxy] [-group_rsp_path <filename>] [-rsp_file <filename>]
[-num_dce_unixd <number>] [cache_lifetime <minutes>]
[-cds_replica_list <list_of_cds_servers>] [-wrap_audit_trail yes | no]
client_components
```

Slim Client Configuration

Notes:

1. The cell administrator's password is not needed when configuring a slim client.
2. Only a DFS client and Security Integration can be configured with a slim client.
3. There are no **admin** configuration steps to perform prior to or after configuring a slim client.

To configure a DCE slim client, perform the following steps as root on the client machine:

To configure a DCE slim client using **smitty**:

1. Start **smitty** with the **mkdceclient** fastpath:

   ```
   smitty mkdceclient
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Clients
   ```

2. Select **local only configuration for this machine**, and press **<Enter>**.
3. Select the **slim_cl** from the list of **CLIENTS to configure**, and press **<Enter>**.
4. Ensure that the **CELL name** field is filled in with the appropriate values.
5. Type the *DCE_hostname* for this machine at the **Machine's DCE HOSTNAME** prompt. If no name is selected, the TCP/IP hostname, including the domain, will be used.
6. Tab to select **yes** or **no** in the **Start components at System restart** field to indicate that the DCE components should or should not be automatically started at system reboot.

7. Tab to select **yes** or **no** in the **Clean up DCE at System restart** field to indicate that **clean_up.dce** should be run at system reboot. This will be done before DCE is restarted.
8. At the **Protocols** prompt, press **<F4>** to list the protocols. Select the protocols that DCE will be configured within, and press **<Enter>**. Once selected, the same protocols must be used for subsequent configurations. If you will be configuring any DFS components on this machine, you must use the **udp** protocols.
9. If the machine is on a separate LAN from any CDS server and cannot send broadcast packets to a CDS server or if you are not sure if the machine is on a separate LAN, type the TCP/IP hostname or IP address of the master security server at the **MASTER SECURITY Server** prompt.
10. **dced** will automatically update the pe_site file with security server information. Type a time to wait between pe_site updates at the **Minutes to wait between pe_site file updates** field.

    ```
       0          (zero) turn automatic update of the pe_site file off
    10 - 1440    10 to 1440 minutes
    ```

    The default is 1440 minutes (24 hours).
11. If the machine is on a separate LAN from any CDS server and cannot send broadcast packets to a CDS server or if you are not sure if the machine is on a separate LAN, type the TCP/IP hostname or IP address of the CDS server at the **CDS Server (If in a separate network)** prompt.
12. If you want to define other CDS servers in the CDS cache, type the TCP/IP hostnames or IP addresses of these CDS servers at the **List of addition CDS Servers** field.
13. Tab to select **yes** or **no** in the **Synchronized Clocks** field to indicate that the local machine's clock should or should not be synchronized with a DTS server.
14. Type the name of the time server you want to use for synchronization at the **Time Server to Synchronize Clocks with** prompt. The time server can be any DTS server.
15. If the machine is on a separate LAN and you want the CDS client to rebroadcast the location of the CDS server, at the **Rebroadcast CDS Server Location** prompt, tab to select **yes**. The default is **no**.
16. Press **<Enter>** to select **Do**.

To configure a DCE slim client from the command line, at the command prompt type:

```
config.dce -cell_name <cell_name>
[-dce_hostname <dce_hostname>] [-sec_master <master_security_server>]
[-cds_server <cds_server>] [-time_server <server id>] [-sync_clocks yes | no]
```

```
[-autostart yes | no] [-clean_autostart yes | no] [-protocol tcp udp]
[-lan_profile <profile>] [-group_rsp_path <filename>] [-rsp_file <filename>]
[-wrap_audit_trail yes | no]
slim_cl
```

At this point, the selected clients are configured on the machine.

Configuring a CDS Client on the Master Security Server

If you configured the master security server and the initial CDS server on the same machine, you can skip this section because a CDS client was configured when you configured the initial CDS server.

Otherwise, to configure a CDS client on the master security server, perform the following steps on the machine that is the master security server:

To configure a CDS client on the master security server using **smitty**:

1. As root, start **smitty** with **mkdceclient** fastpath:

   ```
   smitty mkdceclient
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Clients
   ```

2. Select **full configuration for this machine**, and press **<Enter>**.
3. At the **CLIENTS to configure** prompt, press**<F4>** to list. Select **cds_cl**, and press **<Enter>**.
4. Ensure that the **CELL name** and **SECURITY Server** fields are filled in with the appropriate values.
5. If you are not using the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
6. Ensure that the machine's **DCE HOSTNAME** field is filled in with the appropriate name.
7. Tab to select **yes** or **no** in the **Start components at System restart** field to indicate that the DCE components should or should not be automatically started at system reboot.
8. Tab to select **yes** or **no** in the **Clean up DCE at System restart** field to indicate that **clean_up.dce** should be run at system reboot. This will be done before DCE is restarted.
9. If the master security server is on a separate LAN from the initial CDS server and cannot send broadcast packets to it, type the TCP/IP hostname or IP address of the initial CDS server at the **CDS Server (if in**

a separate network) prompt. (The TCP/IP hostname or IP address is not necessarily the same as the *dce_hostname*.) If you are not sure if it is on a separate LAN, type the TCP/IP hostname or IP address of the initial CDS server at the **CDS Server (if in a separate network)** prompt.

10. Tab to select **yes** or **no** in the **Synchronized Clocks** field to indicate that the local machine's clock should or should not be synchronized with a DTS server.
11. Type the name of the time server you want to use for synchronization at the **Time Server to Synchronize Clocks with** prompt. The time server can be any DTS server.
12. If the cell will contain multiple LANs and require the use of global DTS servers, or if the preferred security replica support will be enabled, type the name of the LAN profile this machine should use at the **LAN PROFILE** field.
13. If this CDS client is on a separate LAN and you want it to rebroadcast the location of the CDS server, at the **Rebroadcast CDS Server Location** prompt tab to select **yes**.
14. All other fields should be filled in because the machine is already configured.
15. Press **<Enter>** to select **Do**.
16. When prompted, type the cell administrator's password.

To configure a CDS client on the master security server from a command line, at the command prompt type:

```
config.dce [-cell_admin <cell_admin>][-cds_server <cds_server>]
cds_cl
```

At this point, a CDS client is configured on the machine.

Further Cell Configuration

After cell initialization is completed, you might have to perform additional configuration tasks on an ongoing basis as changes are made to the cell. For example, you may want a new machine to be added to the cell as a client. Or you may decide to configure a secondary CDS server to provide faster or more reliable access to the namespace.

The following sections provide detailed procedures for performing additional configuration tasks.

Configuring DTS Servers

To configure DTS local or global servers, perform the following steps on each machine designated as a DTS server:

To configure DTS local or global servers using **smitty**:

1. As root, start **smitty** with the **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Servers
   ```

2. Select the **DTS (Distributed Time Service) Server** option. Press **<Enter>**.
3. Ensure that the **CELL name** field is filled in with the appropriate values.
4. If you are not using the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
5. Type the *dce_hostname* for this machine in the **Machine's DCE HOSTNAME** field. If no name is selected, the TCP/IP hostname, including domain, will be used.
6. Tab to select **yes** or **no** in the **Start components at System restart** field to indicate that the DCE components should or should not be automatically started at system reboot.
7. Tab to select **yes** or **no** in the **Clean up DCE at System restart** field to indicate that **clean_up.dce** should be run at system reboot. This will be done before DCE is restarted.
8. In the **Protocol** field, press **<F4>** to list the protocols. Select the protocols with which DCE will be configured, and press **<Enter>**. Once selected, the same protocols must be used for subsequent configurations.
9. If the machine is on a separate LAN from any CDS server and cannot send broadcast packets to it, type the TCP/IP hostname or IP address of the master security server at the **MASTER SECURITY Server** prompt. If you are not sure if it is on a separate LAN, type the TCP/IP hostname or IP address of the master security server at the **MASTER SECURITY Server** prompt.
10. **dced** will automatically update the pe_site file with security server information. Specify a time to wait between pe_site updates at the **Minutes to wait between pe_site file updates** field.

    ```
       0          (zero) turn automatic update of the pe_site file off
    10 - 1440    10 to 1440 minutes
    ```

    The default is 1440 minutes (24 hours).
11. If the machine is on a separate LAN from any CDS server and cannot send broadcast packets to it, type the TCP/IP hostname or IP address of the initial CDS server at the **CDS Server (if in a separate network)**

prompt. If you are not sure if the machine is on a separate LAN, type the TCP/IP hostname or the IP address of the initial CDS server at the **CDS Server (if in a separate network)** prompt.

12. If you want to define other CDS servers in the CDS cache, type the TCP/IP hostnames or IP addresses of these CDS servers at the **List of additional CDS Servers** field.
13. If the cell will contain multiple LANs and require the use of global DTS servers, or if the preferred security replica support will be enabled, type the name of the LAN profile this machine should use at the **LAN PROFILE** field.
14. If the machine is on a separate LAN and you want the CDS client to rebroadcast the location of the CDS server, at the **Rebroadcast CDS Server Location** prompt tab to select **yes**.
15. At the **Type of SERVER** prompt, press **<F4>** to list. Select the appropriate type of server, and press **<Enter>**. Note that a DTS server cannot be configured on the same machine as another DTS server or a DTS client.
16. At the **Type of COURIER** prompt, press**<F4>** to list. Select the appropriate type of courier, and press **<Enter>**.
17. Press **<Enter>** to select **Do**.
18. When prompted, type the cell administrator's password.

To configure DTS local or global servers from a command line, at the command prompt type:

```
config.dce [-courier_role (courier | noncourier | backup)]
[-cell_name <cell_name>] [-cell_admin <cell_admin id>]
[-admin_pwd <admin_password>] [-sec_master <master_security_server>]
[-cds_server <cds_server>] [-lan_profile <profile>]
[-time_server <server id>] [-sync_clocks yes | no]
[-autostart yes | no] [-clean_autostart yes | no] [-protocol tcp udp]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[-wrap_audit_trail yes | no]
dts_local | dts_global
```

If there are less than three time servers configured in the cell, use the following command:

```
dcecp -c dts modify -minservers n
```

where *n* is the number of time servers in the cell. This prevents the logging of a warning message every time the server attempts to sync.

At this point, a DTS server is configured on the machine, along with **dced** (RPC and a security client) and a CDS client which were configured as part of DCE client configuration.

Configuring a DTS Client

To configure a DTS client:

1. As root, start **smitty** with the **mkdceclient** fastpath:

   ```
   smitty mkdceclient
   ```

   or select the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Clients
   ```

2. Select **full Configuration for this machine**, and press **<Enter>**.
3. At the **CLIENTS to configure** prompt, press**<F4>** to list. Select **dts_cl**, and press **Enter**.
4. If you are not using the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
5. All other fields should be automatically filled in with the appropriate values because of a previous configuration on the machine.
6. Press **<Enter>** to select **Do**.
7. When prompted, type the cell administrator's password.

   At this point, a DTS client is configured on the master security server and initial CDS server machines, completing cell initialization.

To configure a DTS client from a command line, at the command prompt type:

```
config.dce [-cell_admin <cell_admin id>] dts_cl
```

Configuring Secondary CDS Servers

After you have configured an initial CDS server, you might want to configure one or more *secondary* CDS servers to provide faster or more reliable access to the namespace.

A Secondary CDS Server allows administrators to create replicas of CDS Directories for backup and availability purposes. When you configure a Secondary CDS Server, a replica of the root directory and its contents is automatically created.

The only child directory below the root that is automatically replicated into the new Secondary CDS Server is the **/.:/subsys/dce/sec** directory. This directory is replicated because it contains the binding information to locate the master security server. This action provides accessibility to the security server even when the initial CDS Server is unavailable. See the *IBM DCE Version 3.2*

for AIX and Solaris: Administration Guide—Core Components for information about CDS replicas and how to create them.

To configure a secondary CDS server, perform the following steps on each machine designated as a secondary CDS server.

Note: If the machine is not already a DCE client, follow the steps outlined in "Configuring DCE Clients" on page 86 to configure it as a DCE client.

To configure a secondary CDS server on a machine already configured as a client using **smitty**:

1. As root, start **smitty** with **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Servers
   ```

2. Select the **CDS (Cell Directory Service) Server** option, and press **<Enter>**.
3. Select the **additional** option, and press **<Enter>**.
4. If you are not using the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
5. Because the machine is already configured as a DCE client, all other fields should be automatically filled in with the appropriate values. Press **<Enter>** to select **Do**.
6. When prompted, type the cell administrator's password.

To configure a secondary CDS server from a command line, at the command prompt type:

```
config.dce [-cell_name <cell_name>] [-cell_admin <cell_admin id>]
[-admin_pwd<admin_password>] [-sec_master <master_security_server>]
[-cds_server <cds_server>] [-lan_profile <profile>]
[clr_house <server id>] [-autostart yes | no] [-clean_autostart yes | no]
[-protocol tcp udp] [-time_server <server id>] [-sync_clocks yes | no]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[-wrap_audit_trail yes | no]
cds_second
```

At this point, **dced** (RPC and a security client), a secondary CDS server, a CDS client, and a DTS client are configured on the machine. When you configure a secondary CDS server, only the **root** and the **/.:/subsys/dce/sec** directories are automatically replicated. Any other directories must be

manually replicated on the secondary CDS servers. See the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide* for information on replicating other directories.

Configuring Security Replica Servers

A security replica server is a read-only copy of the master security server. Advantages of using a security replica server include easing the load on the master security server and preserving the cell in case the master security server becomes disabled.

To configure a security replica server, perform the following steps on each machine designated as a security replica server.

Note: If the machine is not already a DCE client, follow the steps outlined in "Configuring DCE Clients" on page 86 to configure it as a DCE client.

To configure a security replica server using **smitty**:

1. As root, start **smitty** with **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communication Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Servers
   ```

2. Select the **SECURITY Server** option, and press **<Enter>**.
3. Select the **secondary** option, and press **<Enter>**.
4. If you are not using the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
5. If you want to give the security replica a name, type your choice in the **Security Server name** field. If you do not specify a name, the default is the *dce_hostname* of the machine. You should use the default unless you are completely sure that the name you specify is unique throughout the entire cell.
6. Tab to select **yes** or **no** in the **Use LDAP to store security information** field to indicate that the DCE Registry should be stored in LDAP.
7. If LDAP is to be used to store DCE Registry information, enter the names of the LDAP Servers or LDAP Servers and ports to use in the **LDAP SERVER Information List** field.
8. If LDAP is to be used to store DCE Registry information, enter the distinguished name used to authenticate in LDAP in the **LDAP DISTINGUISHED NAME** field.

9. If LDAP is to be used to store DCE Registry information, enter the password for the LDAP Distinguished name in the **LDAP DISTINGUISHED NAME Password** field.
10. If LDAP is to be used to store DCE Registry information, tab to select the LDAP Authentication Method in the **LDAP AUTHENTICATION Method** field. Valid values are none, ssl, gssapi, or cram-md5. The default is **none**.
11. If LDAP is to be used to store DCE Registry information, tab to select **yes** or **no** in the **Use SSL Communication** field to indicate that SSL communication should be used to cummunicate between DCE and LDAP.
12. If LDAP is to be used to store DCE Registry information, type the fully-qualified path for the **LDAP KEYRING** file.
13. Enter the keyring password in the **LDAP KEYRING Password** field if you are using SSL communication. If you do not enter a value, SSL uses the password that is encrypted in the appropriate password stash file.
14. If LDAP is to be used to store DCE Registry information, tab to select **yes** or **no** in the **Store DCE Master Key in LDAP** field to indicate if the DCE Master Key should be stored in LDAP. The default is **no**.
15. If LDAP is to be used to store DCE Registry information, type the fully-qualified path for the **DCE Master Key** file if the DCE Master Key is **NOT** going to be stored in LDAP. The default location is **/opt/dcelocal/var/security/.mkey**.
16. Tab to select **yes** or **no** in the **Use CERTIFICATE based login** field to indicate that PK certificate authentication should be enabled.
17. If PK certificate authentication is to be used, type the fully-qualified path for the file containing the **ENTRUST PROFILE for the security server**.
18. If PK certificate authentication is to be used, type the fully-qualified path for the **ENTRUST INITIALIZATION file**.
19. If PK certificate authentication is to be used, type the **ENTRUST PROFILE password** for the DCE security server.
20. If the machine is already configured as a client, all other fields will be filled in. If the machine is not already configured as a client, specify the TCP/IP hostnames of the Master Security Server and of the CDS Server.
21. Press **<Enter>** to select **Do**.
22. When prompted, type the cell administrator's password.

To configure a security replica server from a command line, at the command prompt type:

```
config.dce [-sec_server_name <security_server>] [-cell_name <cell_name>]
[-cell_admin <cell_admin id>] [-admin_pwd <password>]
[-sec_master <master_security_server>] [-cds_server <cds_server>]
[-autostart yes | no] [-clean_autostart yes | no] [-protocol <tcp udp>]
[-time_server <server_id>] [-sync_clocks yes | no]
```

```
[-certificate_based_login yes | no] [-kdc_profile <kdc_profile>]
[-kdc_ini_file <kdc_ini_file>] [-kdc_passphrase <kdc_passphrase>]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[ldap_auth <none | ssl | cram-md5>] [-ldap_dn <ldap_dn>]
[-ldap_dn_pw <ldap_dn_pw>] [-ldap_keyring <ldap_keyring_file>]
[-ldap_keyring_pw <ldap_keyring_pw>] [-ldap_ssl yes | no]
[-ldap_registry] [-ldap_master_key_in_ldap yes | no]
[-ldap_dce_master_key <dce_master_key_file>]
[-ldap_server <ldap_server | ldap_server:port_number>]
sec_rsp
```

At this point, **dced** (RPC and a security client), a security replica, and a CDS client are configured on the machine.

Configuring the Global Directory Agent

The Global Directory Agent (GDA) allows intercell communication by locating a foreign cell which has been registered into the Domain Naming System (DNS) global directory service. Only one GDA is required to be configured within the cell to allow intercell communication, but more can be configured to increase availability.

To configure the GDA on a machine, perform the following steps on the machine.

Note: If the machine is not already a DCE client, follow the steps outlined in "Configuring DCE Clients" on page 86.

To configure GDA using **smitty**:

1. As root, start **smitty** with **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Servers
   ```

2. Select the **GDA (Global Directory Agent)** option, and press **<Enter>**.
3. If you are not using the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
4. Enter the names of the LDAP Servers or LDAP Servers and ports to use in the **The LDAP server host id with optional ':port_number'.** field.
5. Because the machine is already configured as a DCE client, all other fields should be automatically filled in with the appropriate values. Press **<Enter>** to select **Do**.
6. When prompted, type the cell administrator's password.

To configure GDA from a command prompt, at the command line type:

```
config.dce [-cell_name <cell_name>] [-admin_pwd<admin_password>]
[-cell_admin <cell_admin id>] [-sec_master <master_security_server>]
[-cds_server <cds_server>] [-lan_profile <profile>]
[-ldap_server <ldap_server | ldap_server:port_number>]
[-time_server <server id>] [-sync_clocks yes | no]
[-autostart yes | no] [-clean_autostart yes | no] [-protocol tcp udp]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[-wrap_audit_trail yes | no]
gda_srv
```

At this point, the GDA is configured on the machine. To enable intercell communication, see the information on the intercell environment in the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*. Also, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components* for information on registering a cell globally.

Configuring EMS Servers

To configure an EMS server perform the following steps.

Note: If the machine is not already a DCE client, follow the steps outlined in "Configuring DCE Clients" on page 86.

To configure an EMS server using **smitty**:

1. As root, start **smitty** with the **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   or select the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Servers
   ```

2. Select the **EMS Server** option, and press **<Enter>**.
3. Because the machine is already configured as a DCE client, all other fields should be automatically filled in with the appropriate values. You will not be prompted for the cell administrator's password when the DCE client is already configured. Press **<Enter>** to select **Do**.

To configure an EMS server from a command line, at the command prompt type:

```
config.dce [-cell_name <cell_name>] [-admin_pwd<admin_password>]
[-cell_admin <cell_admin id>] [-sec_master <master_security_server>]
[-cds_server <cds_server>] [-lan_profile <profile>]
[-time_server <server id>] [-sync_clocks yes | no]
```

```
[-autostart yes | no] [-clean_autostart yes | no] [-protocol tcp udp]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[-wrap_audit_trail yes | no]
ems_srv
```

At this point, an EMS server is configured on the machine, along with **dced** (RPC and a security client) and a CDS client which were configured as part of DCE client configuration.

Configuring SNMP Servers

To configure an SNMP server perform the following steps.

To configure an SNMP server using **smitty**:

1. As root, start **smitty** with the **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   or select the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Servers
   ```

2. Select the **SNMP Server** option, and press **<Enter>**.
3. Tab to select **yes** or **no** in the **Start components at System restart** field to indicate that the DCE components should or should not be automatically started at system reboot.
4. Tab to select **yes** or **no** in the **Clean up DCE at System restart** field to indicate that **clean_up.dce** should be run at system reboot. This will be done before DCE is restarted.
5. Press **<Enter>** to select **Do**.

To configure an SNMP server from a command line, at the command prompt type:

```
config.dce
[-autostart yes | no] [-clean_autostart yes | no]
snmp_srv
```

Configuring DCE 3.2 for AIX Security Integration

Use the following steps to configure the dceunixd daemon.

Note: If the machine is not already a DCE client, follow the steps outlined in "Configuring DCE Clients" on page 86.

To configure a system for security integration operations using **smitty**:

1. As root, start **smitty** with the mkdcesrv fastpath:

```
smitty mkdcesrv
```

or select the following sequence of **smitty** menu options:

```
1. Communications Applications and Services
2. DCE (Distributed Computing Environment)
3. Configure DCE/DFS
4. Configure DCE/DFS Servers
```

2. Select the **DCE UNIXD Server** option, and press **<Enter>**.
3. If you are not using the default cell_admin, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
4. Enter the number of dceunixd processes you want to run at the **Number of instances of dce unixd processes** field.
5. Enter the number of minutes that the dceunixd cache should be kept at the **Cache life time in minutes** field.
6. Because the machine is already configured as a DCE client, all other fields should be automatically filled in with the appropriate values. Press **<Enter>** to select **Do**.

To configure a system for security integration operations from a command line, at the command prompt type:

```
config.dce [-cache_lifetime <minutes>][-num_dce_unixd <num>]
dce_unixd
```

At this point, a **dceunixd** server is configured on the machine, along with **dced** (RPC and a security client) and a CDS client which were configured as part of DCE client configuration. To set up the machine to use DCE security integrated login, see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components* for complete details.

Configuring Audit Servers

To configure an Audit server perform the following steps.

Notes:

1. If the machine is not already a DCE client, follow the steps outlined in "Configuring DCE Clients" on page 86.
2. To allow the server to use auditing, you must have the environment variable set DCEAUDITON=1. The easiest way to accomplish this is to configure audit and then stop and restart the servers, making sure the DCEAUDITON environment variable is set before you start.

To configure an Audit server using **smitty**:

1. As root, start **smitty** with the **mkdcesrv** fastpath:

```
smitty mkdcesrv
```

or select the following sequence of **smitty** menu options:

```
1. Communications Applications and Services
2. DCE (Distributed Computing Environment)
3. Configure DCE/DFS
4. Configure DCE/DFS Servers
```

2. Select the **Audit Daemon** option, and press **<Enter>**.
3. Tab to select yes or no in the Wrap Audit Trail field to indicate if the audit trail should wrap.
4. Because the machine is already configured as a DCE client, all other fields should be automatically filled in with the appropriate values. You will not be prompted for the cell administrator's password when the DCE client is already configured. Press **<Enter>** to select **Do**.

To configure an Audit server from a command line, at the command prompt type:

```
config.dce [-cell_name <cell_name>]
[-sec_master <master_security_server>] [-cds_server <cds_server>]
[-lan_profile <profile>][-autostart yes | no] [-clean_autostart yes | no]
[-protocol tcp udp][-time_server <server id>] [-sync_clocks yes | no]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[-wrap_audit_trail yes | no]
audit
```

At this point, an Audit server is configured on the machine, along with **dced** (RPC and a security client) and a CDS client which were configured as part of DCE client configuration.

Configuring Password Strength Servers

To configure a Password Strength server on a machine, perform the following steps on the machine.

Note: If the machine is not already a DCE client, follow the steps outlined in "Configuring DCE Clients" on page 86.

To configure a Password Strength server using **smitty**:

1. As root, start **smitty** with the **mkdcesrv** fastpath:

```
smitty mkdcesrv
```

Or select the following sequence of **smitty** menu options:

1. **Communications Applications and Services**
2. **DCE (Distributed Computing Environment)**
3. **Configure DCE/DFS**
4. **Configure DCE/DFS Servers**

2. Select the **Password Strength Server** option, and press **<Enter>**.
3. Select either the **Basic Server** option or **Enhanced IBM Server** option, and press **<Enter>**.
4. If you are not using the default **cell_admin**, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
5. If you are not using the default password strength server arguments, type the arguments at the **Password strength server argument** prompt.
6. If you are not using the default password strength server name, type the password strength server command, including the full path, at the **Password strength server daemon** prompt.
7. If you are not using the default password strength server principal, type the password strength principal at the **Principal ID for Password strength server** prompt.
8. Tab to select **yes** or **no** in the **Allow user-defined libraries** field to indicate that this server should or should not be allowed to access user-defined libraries for additional rule enforcement. (This option will only be available if the Enhanced IBM Server was selected in step 3.)
9. Because the machine is already configured as a DCE client, all other fields should be automatically filled in with the appropriate values. Press **<Enter>** to select **Do**.
10. When prompted, type the cell administrator's password.

To configure a Password Strength server from a command line, at the command prompt type:

```
config.dce [-cell_name <cell_name>] [-cell_admin <cell_admin id>]
[-admin_pwd <admin_password>] [-sec_master <master_security_server>]
[-cds_server <cds_server>][-lan_profile <profile>]
[-pwdstr_arg <command line args>] [-pwdstr_cmd <server_name>]
[-pwdstr_principal <password strength principal id>]
[-autostart yes | no] [-clean_autostart yes | no] [-protocol tcp udp]
[-time_server <server id>] [-sync_clocks yes | no]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[-wrap_audit_trail yes | no]
pw_strength_srv
```

Notes:

1. If you apply more than one command line argument to the **-pwdstr_arg** option, the arguments must be enclosed by double quotation marks (″). For example:

   ```
   -pwdstr_arg "-v -d"
   ```

2. If you want to specify a password strength principal other than pwd_strengthd, specify both the**-pwdstr_principal** option and the **-pwdstr_arg -server_princ** option. For example:

```
-pwdstr_principal pwd_server
-pwdstr_arg "-server_princ pwd_server"
```

At this point, a password strength server is configured on the machine, along with **dced** (RPC and a security client) and a CDS client which were configured as part of DCE client configuration.

Configuring the Name Service Interface Daemon (NSID)

To configure NSID on a machine, perform the following steps on the machine.

Note: If the machine is not already a DCE client, follow the steps outlined in "Configuring DCE Clients" on page 86.

To configure NSID using **smitty**:

1. As root, start **smitty** with the **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   Or select the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Servers
   ```

2. Select the **NSI (Name Service Interface) Daemon** option, and press **<Enter>**.
3. If you are not using the default **cell_admin**, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
4. Because the machine is already configured as a DCE client, all other fields should be automatically filled in with the appropriate values. Press **<Enter>** to select **Do**.
5. When prompted, type the cell administrator's password.

To configure NSID from a command line, at the command prompt type:

```
config.dce [-cell_name <cell_name>] [-cell_admin <cell_admin id>]
[-admin_pwd <admin_password>] [-sec_master <master_security_server>]
[-cds_server <cds_server>] [-lan_profile <profile>]
[-time_server <server id>] [-sync_clocks yes | no]
[-autostart yes | no] [-clean_autostart yes | no] [protocol tcp udp]
[-group_rsp_path <filename>] [-rsp_file <filename>]
[-nsid_pwd <nsid_password>] [-wrap_audit_trail yes | no]
nsid
```

At this point, NSID is configured on the machine, along with **dced** (RPC and a security client) and a CDS client which were configured as part of DCE client configuration.

Configuring an Identity Mapping Server

To configure an Identity Mapping server, perform the following steps on the machine using **smitty**:

Note: If the machine is not already configured as a security server, follow the steps outlined in "Configuring the Master Security Server" on page 81 or "Configuring Security Replica Servers" on page 99.

1. As root, start **smitty** with the **mkdcesrv** fastpath:

   ```
   smitty mkdcesrv
   ```

   or perform the following sequence of **smitty** menu options:

   ```
   1. Communications Applications and Services
   2. DCE (Distributed Computing Environment)
   3. Configure DCE/DFS
   4. Configure DCE/DFS Servers
   ```

2. Select the **Identity Mapping Server** option, and press **<Enter>**.
3. If you do not want to use the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
4. Tab to select **yes** or **no** in the **Start components at System restart** field to indicate that the DCE components should or should not be automatically started at system reboot.
5. Tab to select **yes** or **no** in the **Clean up DCE at System restart** field to indicate that clean_up.dce should be run at system reboot. This will be done before DCE is restarted.
6. At the **ENTRUST PROFILE for the Security server** prompt, type the Entrust profile name, including the full path name.
7. At the **ENTRUST INITIALIZATION file** prompt, type the Entrust Initialization file name, including the full path.
8. At the **ENTRUST PROFILE Password** prompt, type the password associated with the Entrust profile for the DCE Security Server.
9. Press **<Enter>** to select **Do**.
10. When prompted, type the cell administrator's password.

To configure an Identity Mapping Server from a command line, the Identity Mapping server must be configured on the same machine as either the master security server or a security replica server. At the prompt type:

```
config.dce [-autostart yes | no] [-clean_autostart yes | no] [kdc_profile <profile file>
[-kdc_ini_file <Entrust initialization file>] [-kdc_passphrase <Entrust profile password
idms_srv
```

When prompted, type the cell administrator's password.

At this point, an Identity Mapping server, security server (master or replica), **dced** (RPC and a security client) and a CDS client are configured on the machine.

Note: Modify the registry version by issuing the following command at a command line:

```
dcecp -c registry modify -version {secd.dce.1.2.2a}
```

Configuring DCE Web Secure for AIX

DCE Web Secure must be installed and configured on a workstation that has a DCE client and an iPlanet FastTrack 4.1, iPlanet Enterprise 4.0, or iPlanet Enterprise 4.1 Web server. DCE Web Secure also supports the Web servers supported by DCE 3.1. DCE Web Secure for AIX can be configured using SMIT.

Configuring DCE Web Secure Using SMIT

From the main SMIT panels:

1. Select **Communications Applications and Services**.
2. Select **DCE (Distributed Computing Environment)**.
3. Select **Configure / Unconfigure DCE Web**.
4. Select **Configure DCE Web**.
5. Type the root directory where your Netscape server is installed in the **Netscape Directory** field and press **<Enter>**.
6. At the **Configure DCE Web** panel:
   - Type the ID of the Netscape server in the **Netscape server ID** field.
   - Type a *userid* in the **User ID** field.
   - Select **All** in the **Components to configure** field. **All** is the default to configure DCE Web Secure and DCE Web Administration.
7. Select **OK** and press **<Enter>** to begin configuration.

The *netscape server root directory* is the root directory where your Netscape server is installed.

The *Netscape Server ID* identifies the computer name of the machine where the Netscape Web server was installed. This is the *hostname* field in the View Server Settings window in the iPlanet Web Server Administration Server GUI.

It is also the *ServerName* identified in the **magnus.conf** file. If you are unsure about the value to enter, press F4 for a list of options.

The *userid* is the operating system user account name for the Netscape server to run under. The *userid* cannot be *nobody*.

The component can be **secure**, **admin**, or **all**. **secure** configures DCE Web Secure. **admin** and **all** configure both DCE Web Secure and DCE Web Administration. For more information see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.

Configuring DCE Web Secure from a Command Line

To configure DCE Web Secure for AIX or DCE Web Secure and DCE Web Administration for AIX from the command line, type:

```
mkdceweb -n <netscape_dir>  -s <netscape_id>  -i <user_id> -t <component>
```

- The <netscape_dir> is the root directory where your Netscape server product is installed.
- The <netscape_id> identifies the name of the Web server. This value comes from the **Server Identifier** field specified by the administrator through the Netscape Administration Server when the server was installed.
- The <user_id> is the operating system user account name under which the Netscape server should run. The <user_id> cannot be *nobody*, and it is recommended that the Web server not be run as *root*.
- The <component> indicates which components to configure:
  - **secure** configures DCE Web secure only.
  - **admin** configures DCE Web Administration and DCE Web Secure, since DCE Web Secure is a prerequisite of DCE Web Administration.
  - **all** configures DCE Web Secure and DCE Web Administration.

Note: For some Asian locales, you can not start a Netscape Web server, such as DCE Web Secure, from the command line in an aixterm. If you plan to configure or start DCE Web Secure from the command line, use a dtterm, not an aixterm. Alternatively, use the Netscape Server Administration page rather than the command line to manage DCE Web Secure.

Verifying Configuration of DCE Web Secure

The best way to verify whether or not your configuration was successful, besides the absence of error messages during configuration, is to use the features of DCE Web Secure through your Web browser. To do this, type the following Web address in your Web browser:

```
http://netscape_id[:port]/dceweb
```

netscape_id is the name of the Web server specified when you configured DCE Web Secure. The port is optional if you are using port 80, the default port.

Accessing a CGI using DCE Credentials

With DCE Web Secure installed and configured, you can provide DCE credentials to Common Gateway Interface (CGI) programs. This functionality can be especially useful if you want to run a shell script or tcl script that needs DCE credentials from your Web browser.

As an example, place the following shell script in a file called testcgi.sh in /opt/dcelocal/web/admin/cgi-bin. Ensure the shell script has execute permission for the operating system userid under which the Web server is running.

You may need to change the Netscape Browser Font to code page 850. To do this:

1. Click **Edit→Preferences→Appearance→Fonts**. Select **User-Defined** for the Encoding and **ibm-850**for the fonts.
2. Click**View→Character Set**. Select **User Defined**.

```
#!/bin/sh
 # testcgi.sh Test CGI program to show DCE credentials.
 echo "Content-type: text/html"
 echo ""
 echo "<html>"
 echo "<head>"
 echo "<title>Test CGI program to show DCE credentials</title>"
 echo "</head>"
 echo "<body>"
 echo "<h1>Test CGI program to show DCE credentials</h1>"
 echo "<p><h3>CGI is running under the following DCE credentials:</h3>"
 echo "<pre>"
 klist | grep "Global Principal"
 echo "</pre>"
 echo "</body>"
 echo "</html>"
```

When run, the CGI will show you which DCE credentials you are using. For example, if you stored the file in **/opt/dcelocal/web/admin/cgi-bin/testcgi.sh** and you setup your Web server to allow CGI programs to run from **/opt/dcelocal/web/admin/cgi-bin**, you can run this CGI from the following Web address:

```
http://<<server-name>/dceweb/cgi-bin/testcgi.sh
```

To ensure that the CGI runs with DCE credentials, check that unauthenticated access is not turned on for the **/opt/dcelocal/web/admin/cgi-bin** path in your Web server configuration.

Running a CGI Program in the Default Path

The Web Secure Configuration sets the default CGI path in the **obj.conf** file. The default path is **/opt/dcelocal/web/admin/cgi-bin** (map to **/dceweb/cgi-bin**), so CGI scripts and applications could be copied to that directory.

To run a CGI application URL:

```
http://server_name:port/dceweb/cgi-bin/cgi_script_name
```

server_name
The host name for the Web Server

port The port number which Web Server is in. This paramater is optional if the port number is *80*.

cgi_script_name
is the file name of the CGI script or application

The user can specify a different path to store the CGI scripts and applications, but the user will have to change the **obj.conf** file manually or through the iPlanet Web Server Administration Server GUI.

Changing the Active Locale for DCE Web Administration

First stop Web Secure, change its locale, and restart it.

For example, from the command line, you might use commands similar to these:

```
>/usr/netscape/suitespot/httpd-<server_name>/stop
>export LC_ALL=<locale>
>/usr/netscape/suitespot/httpd-<server_name>/start
```

Next, stop and start the Netscape Browser and access DCE Web Administration.

Unconfiguring DCE Components

Occasionally, certain situations require that you unconfigure (or remove configuration and database files for) a particular DCE component from a machine. For example, if you want to reconfigure a particular component with new parameters, you must unconfigure it to remove the existing configuration before setting up the new configuration. Or, if configuration of a component failed, leaving it only partially configured, you must remove the partial configuration before attempting configuration again.

Other situations require that you unconfigure an entire machine (that is, unconfigure all DCE components from the machine). For example, if you want

to transfer a machine from one cell to another, you must remove the configurations for the old cell from the machine before setting up the configurations for the new cell.

In rare cases, you might want to unconfigure an entire cell. If you unconfigure a cell, you should also unregister its name from the global namespace.

Attention: After you unconfigure a secondary CDS server (**unconfig.dce cds_second**), you must wait two hours before you reconfigure a secondary CDS server with the same name. The master CDS server refreshes its identity at two-hour intervals.

The following section provides more information on unconfiguring DCE components.

Considerations Before Unconfiguring

Note: The information in this section pertains to legacy DCE. For information on this topic related to the DCE Security Registry and LDAP Integration feature, refer to the *IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide*.

You should exercise caution in unconfiguring DCE components, especially if you are removing components which perform services required by other components. Unconfiguring a component will partially or completely disable other components which are dependent upon it.

Attention: In the event that you must reconfigure a cell and you are running DFS in your environment, refer to your DFS documentation supporting IBM DCE 3.2 for considerations before reconfiguring a cell.

There are special cases which you should take into consideration when unconfiguring DCE components:

- The master security server and the CDS server that contain the master replica of the **/.:** directory are the basis of any cell. If you unconfigure one or both of these servers, you have to unconfigure and rebuild your entire cell.
- To unconfigure the Master Security Server, you must use the **local** option.
- To unconfigure a CDS server that has a master replica of any directory, you must use the **local** option.

  Note: See the *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components* for more information on changing the location of a directory's master replica.

When you unconfigure DCE components on a machine, two types of operations are performed:

- Local operations (updating configuration files and stopping daemons)
- Administrative operations (updating the security registry, the CDS namespace)

Just as configuration is separated into **admin** and **local** portions, so is most of unconfiguration. The exceptions are the master security server (**sec_srv**) and any CDS server (**cds_srv** or **cds_second**) that contains a master replica of a directory in one of its clearinghouses.

When you unconfigure DCE components on a machine, if all the local operations can be undone, the machine itself is considered to be unconfigured. However, if attempts to undo administrative operations fail, the machine is not fully unconfigured from the cell; entries for the machine might still exist in the CDS namespace or registry databases. On a full unconfiguration if attempts to undo administrative operations fail, a list of the failed operations is printed to **/opt/dcelocal/etc/cfgdce.log** so you can manually perform these operations and remove references to the machine from the namespace and registry databases. From another machine configured in your cell, you can run **admin** unconfiguration for operations that failed so that you can clean up the DCE registry database and the namespace.

Refer to the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* for complete information on the DCE commands referenced above.

Split Unconfiguration

Sometimes it is beneficial to use a feature known as the *split unconfiguration of clients*, which allows the root user to perform the unconfiguration steps on the local machine while the cell administrator cleans up the rest of the cell. A local unconfiguration is useful in the following situations:

- If the cell for which a machine is configured is inaccessible or you do not have the password for that cell administrator's account, you need only to remove the local configuration files from the machine to reconfigure it for a new cell.
- If the configuration of a machine is so broken that it cannot reach the security server to be authenticated to perform remote operations, you can limit unconfiguration to local items.
- If you are unconfiguring the master security server, you must limit unconfiguration to local items.
- If you are unconfiguring a CDS server that contains the master replica of a directory, you must limit unconfiguration to local items.
- If you are unconfiguring a slim client, only local unconfiguration steps are necessary.

The cell administrator should run the **admin** portion of unconfiguration from a machine in the cell to complete the unconfiguration process. A full client

that has been locally unconfigured cannot be configured back into the cell until the admin portion of unconfiguration has been done.

Steps for Unconfiguring DCE

To unconfigure one or more DCE components from a machine, perform the following steps:

- To unconfigure one or more DCE components using SMIT:
  1. As root, start SMIT with the **unconfig.dce** fastpath:

     ```
     smitty rmdce
     ```

     or select the following sequence of SMIT menu options:

     ```
     1. Communications Applications and Services
     2. DCE (Distributed Computing Environment)
     3. Unconfigure DCE/DFS
     ```

  2. At the **Type of Unconfiguration** select box, select one of the following:
     - **full unconfiguration for this machine**
     - **local only unconfiguration for this machine**
     - **admin only unconfiguration for another machine**
  3. At the **COMPONENTS to Remove** panel type or select from the pull-down list, the components that you want to remove.

     For the **admin only unconfiguration**, type the *dce_hostname* of the machine for which you are unconfiguring components in the **Client Machine's DCE HOSTNAME** field.

     For the **full unconfiguration** and **local unconfiguration**, the **Remove DEPENDENT Components?** field defaults to **No**. You should change this field to **Yes** only if you have selected a component and are *sure* that you want to unconfigure every component that depends on the presence of the component you selected. For example, all components depend on the presence of **dced**. Therefore, if you select **dced** as the only client to unconfigure and change **Remove DEPENDENT Components?** to **Yes**, the result will be the same as if you had selected **All** for **COMPONENTS to Remove**.

     Note: If you are unconfiguring a Password Strength server, you must type its ID in the **Principal ID for Password Strength Server** field.
  4. If you are not using the default *cell_admin*, type the name of the cell administrator's account at the **Cell ADMINISTRATOR's account** prompt.
  5. For the **full unconfiguration** and the **local only unconfiguration**, the **OVERRIDE Dependency Checking?** field defaults to **No**. You should change this field to **Yes** only if you are *sure* that you want to

unconfigure a component without unconfiguring other components that are dependent on it. For example, if you unconfigure RPC but leave **sec_cl** and **cds_cl** configured, these two will not be able to function properly.

6. Select **Do**.
7. If prompted, type the cell administrator's password for the **full unconfiguration** and the **admin unconfiguration**.

- To unconfigure one or more DCE components from a command line, at the command prompt type:

```
unconfig.dce -config_type full
[-cell_admin <cell_admin id>] [-dependents]
[-force] [-pwdstr_principal <password_strength_principal id>]components
```

Unconfiguring DCE Web Secure

You **must** unconfigure DCE Web Secure before uninstalling it.

DCE Web Secure can be unconfigured using SMIT or from a command line.

To Unconfigure DCE Web Secure Using SMIT:

From the main SMIT panels:

1. Select **Communications Applications and Services**.
2. Select **DCE (Distributed Computing Environment)**.
3. Select **Configure / Unconfigure DCE Web** .
4. Select **Unconfigure DCE Web**.
5. At the **Netscape Directory** panel:
   - Type the root directory where your Netscape server is installed in the **Netscape Directory** field and press <Enter>.
6. At the **Unconfigure DCE Web** panel:
   - Type the ID of the Netscape server in the **Netscape server ID** field.
   - Type a userid in the **User ID** field.
   - Select **All** in the **Components to unconfigure** field. **All** is the default to unconfigure DCE Web Secure and DCE Web Administration.
   - Select either **Yes** or **No** in the **Remove Archived Configuration files** field. **No** is the default.
7. Select **OK** and press **<Enter>** to begin unconfiguration.

For more information see the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*.

Unconfiguring DCE Web Secure from a Command Line

To unconfigure DCE Web Secure for AIX from the command line, type:

```
rmdceweb -n <netscape_dir> -s <netscape_id> -t <component>
```

- The <netscape_dir> is the root directory where your Netscape server product is installed.
- The <netscape_id> identifies the name of the Web server. This value comes from the **Server Identifier** field specified by the administrator through the Netscape Administration Server when the server was installed.
- The <component> indicates which components to unconfigure:
  - **secure** unconfigures DCE Web Secure and DCE Web Administration, since DCE Web Secure is a prerequisite of DCE Web Administration.
  - **admin** unconfigures DCE Web Administration.
  - **all** unconfigures DCE Web Secure and DCE Web Administration.

Chapter 5. Starting and Stopping DCE 3.2 for AIX

Starting DCE Daemons

You can use either the command line or the SMIT interface to start DCE daemons.

Using the Command Line to Start Daemons

The **start.dce** command starts DCE daemons for configured DCE components. Before starting DCE daemons, you must be logged in as root.

To start all daemons for configured DCE components, type the following at the command line:

```
start.dce all
```

To start all daemons for configured DCE components, type either of the following commands at the command line:

```
start.dce core
start.dce
```

To start specific configured components, add the component name, such as **cds_srv**, to the command:

```
start.dce cds_srv
```

Note: If the master security server and the Initial CDS server are on different machines and both have been stopped, use the following steps to restart DCE:

Machine 1
(rpc, sec_cl, sec_srv, cds_cl, and any other dce components)

Machine 2
(rpc, sec_cl, cds_srv, cds_cl, and any other dce components)

1. Machine 1: **start.dce rpc sec_cl sec_srv**
2. Machine 2: **start.dce rpc sec_cl cds_cl cds_srv**
3. Machine 1: **start.dce all**
4. Machine 2: **start.dce all**

Using SMIT to Start DCE Immediately and at System Restart

You can run **start.dce** now to start all configured DCE and DFS daemons. You can also run **start.dce** at system restart if the appropriate entries are in **/etc/inittab**.

Note: For compatibility and to be consistent with other AIX commands, the **rc.(_)** commands are put into **/etc/inittab**. **rc.dce** will invoke **start.dce**. Use SMIT to add or delete these entries.

1. As root, start SMIT:

   ```
   smitty mkdceitab
   ```

2. Tab to select **yes** or **n**o in the **Start DCE now** field. Selecting **yes** will cause **start.dce** to be run immediately. The default is **yes**.
3. Tab to select **yes** or **no** in the **Start DCE at system restart** field. Selecting **yes** will cause **rc.dce** to be put into **/etc/inittab**, selecting **no** will remove **rc.dce** from **/etc/inittab**. The default is **no**.
4. Tab to select **yes** or **no** in the **Clean up DCE files at system reboot** field. Selecting **yes** will cause **clean_up.dce** to put into **/etc/inittab**, selecting **no** will remove **clean_up.dce** from **/etc/inittab**. The default is **no**.
5. Click **Do**.

Note: Options on the **config.dce** command allow you to turn autostart on or off.

Changing Arguments Passed to DCE Daemons at Restart

The arguments that are passed to the DCE daemons during startup can be changed by the user.

Command line parameters for daemons can be changed by editing the file **/opt/dcelocal/etc/cfgarg.dat**. Be sure to back up the file before editing.

Each daemon that is configured on the system will have an entry in **cfgarg.dat**. Each entry will have the following format:

```
daemon_name: command line options
```

For example, the entry for dced daemon in the **cfgarg.dat** file might look like the following:

```
dced: -b -t 1440
```

If a specific daemon starts without arguments, nothing will be listed after the colon.

Following is an example of the **cfgarg.dat** file for a machine that is configured as a Master Security server, CDS Initial Server, Security client, and CDS client:

```
dced: -b -t 1440
secd:
cdsadv:
cdsd:
```

If the arguments are changed in the **cfgarg.dat** file while DCE is running, DCE must be stopped and restarted for the new arguments to take effect.

Note: Please refer to the *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference* for a complete listing of arguments for each DCE daemon.

Stopping DCE Daemons

The **stop.dce** command stops DCE daemons for configured DCE components. To stop DCE daemons, you must be logged in as root.

To stop all daemons for configured DCE and DFS components, type one of the following at the command line:

```
stop.dce core
stop.dce
stop.dce all
```

To stop specific daemons for configured DCE components, add the daemon's name to the **stop.dce** command and type the following at the command line:

```
stop.dce dts_cl
```

Chapter 6. Obtaining Additional Information

This chapter describes the sources of information that can be useful when you are using DCE 3.2 for AIX.

Books

The DCE 3.2 for AIX library contains a printed copy and an online version of the *IBM DCE Version 3.2 for AIX: Quick Beginnings* and a printed copy of the *IBM DCE Version 3.2 for AIX: Release Notes*. All other supporting product documentation is provided only in online format.

Online Information

Extensive online documentation is shipped as part of the DCE for AIX product. The IBM DCE 3.2 for AIX online documentation is provided in three file formats:

- HTML files that are viewable from any frame-enabled Web browser, such as Netscape Navigator.
- PDF files that are viewable with a PDF viewer, such as Adobe Acrobat Reader.
- Plain text files that are viewable with any text editor, such as vi.

Online Books

The following IBM DCE books are available online:

- *IBM DCE Version 3.2 for AIX: Quick Beginnings*
- *IBM DCE Version 3.2 for AIX and Solaris: Introduction to DCE*
- *IBM DCE Version 3.2 for AIX and Solaris: Problem Determination Guide*
- *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Introduction*
- *IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*
- *IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*
- *IBM DCE Version 3.2 for AIX: High Availability Cluster Multi-Processing Guide for DCE and DFS*
- *IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide*
- *IBM DCE Version 3.2 for AIX and Solaris: Application Development Guide—Introduction and Style Guide*
- *IBM DCE Version 3.2 for AIX and Solaris: Application Development Guide—Core Components*

- *IBM DCE Version 3.2 for AIX and Solaris: Application Development Guide—Directory Services*
- *IBM DCE Version 3.2 for AIX and Solaris: Application Development Reference*

Help Files

DCE 3.2 for AIX provides assistance for system management tasks in the form of SMIT helps. It also provides HTML helps for the DCE Administration Graphical User Interface (GUI).

Print and Order Books

IBM DCE Publications

In addition to the hardcopy editions of the *IBM DCE Version 3.2 for AIX: Quick Beginnings* and *IBM DCE Version 3.2 for AIX: Release Notes*, IBM supplies PDF files on the CD-ROM for each of the online DCE 3.2 documents for those customers who want the option of having printed documentation.

Other Publications

Although not written specifically for AIX products, the following O'Reilly books might be useful:

- Hu, Wei. *DCE Security Programming*, 1st. ed. Sebastopol, CA: O'Reilly & Associates, 1994.
- Rosenberry, Ward. *Understanding DCE*, 2nd. ed. Sebastopol, CA: O'Reilly & Associates, 1993.
- Shirley, John. *Guide to Writing DCE Applications*, 2nd. ed. Sebastopol, CA: O'Reilly & Associates, 1994.

Using DCE 3.2 for AIX Documentation

The DCE 3.2 for AIX product includes user, administration, and application development documentation that is accessible online.

See "Appendix A. Online Documentation" on page 127 for information about the filesets that must be installed to access the DCE for AIX online documentation.

Note: Not all books are translated for each language. Non-translated books will view in English.

The start_dcedoc program

You can use the **start_dcedoc** program to view the DCE online documentation. The **start_dcedoc** program defaults to the appropriate viewer for your interface based on your **$DISPLAY** environment variable. You can also use flags to specify which viewer you want **start_dcedoc** to start. The flags are **-a** for an ASCII interface and **-g** for a graphics interface. The **-a** flag enables you

to view English-only documentation in plain text format. To view the documentation in HTML format for any language, use the **-g** and **-lang** flags as follows:

```
start_dcedoc -g -lang locale
```

where *locale* can be substituted with one of the following locale names:

en_US English locale

ja_JP Japanese (EUC) locale

Ja_JP Japanese (PC Kanji) locale

JA_JP Japanese (UTF-8) locale

ko_KR
Korean locale

KO_KR
Korean (UTF-8) locale

pt_BR Brazilian Portuguese locale

PT_BR
Brazilian Portuguese (UTF-8) locale

zh_TW
Traditional Chinese (EUC) locale

Zh_TW
Traditional Chinese (Big5) locale

ZH_TW
Traditional Chinese (UTF-8) locale

Note: The **-g** flag starts the Netscape Navigator browser. If you don't have Netscape Navigator installed, **start_dcedoc** will fail. For further information, see "Viewing the HTML Documentation using a Web Browser".

Viewing Plain Text Files

To view the DCE online documentation in plain text format:

1. Change to the appropriate directory **/usr/lpp/dcedoc/3270/en_US**
2. Browse booklist to identify the correct document
3. View the selected document using any plain text editor, such as vi

Note: en_US can be substituted with one of the preceding locale names.

Viewing the HTML Documentation using a Web Browser

Users with graphic interfaces can use a Web browser such as the Netscape Navigator browser, which is included with the AIX operating system, to read

the DCE documentation HTML files. The Netscape Navigator browser provides hypertext linking, navigation utilities, a hypertext index, graphical display of artwork, search and print facilities, a bookmark function, and an NLS-enabled online help utility. See the AIX documentation for information on installing the Netscape Navigator browser.

If you have installed the documentation files locally, use your Web browser to view the DCE HTML documentation by opening the file:

```
/usr/lpp/dcedoc/html/en_US/index.html
```

Note: en_US can be substituted with one of the preceding locale names.

If you have installed the documentation files and DCE Web Secure is installed and configured, use your Web browser remotely to view the DCE HTML documentation at the following Web address:

```
http://<servername>/dcedoc/en_US
```

where <*servername*> is the name of the server where you installed and configured DCE Web Secure.

Note: en_US can be substituted with one of the preceding locale names.

Printing the PDF Books

If you prefer hardcopy documentation, a set of PDF files is included on the product CD-ROM. You can print these books directly from the CD-ROM. Go to the location **/usr/lpp/dcedoc/pdf/en_US/** and select the PDF file that you want to send to your printer. See "Appendix A. Online Documentation" on page 127 for a listing of the publications and their file prefixes.

Note: en_US can be substituted with one of the preceding locale names.

Appendix A. Online Documentation

The following table identifies the documents by file prefix:

Prefix	Title
admingd	*IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Core Components*
admintro	*IBM DCE Version 3.2 for AIX and Solaris: Administration Guide—Introduction*
aixquick	*IBM DCE Version 3.2 for AIX: Quick Beginnings*
apgstyle	*IBM DCE Version 3.2 for AIX and Solaris: Application Development Guide—Introduction and Style Guide*
appdev	*IBM DCE Version 3.2 for AIX and Solaris: Application Development Guide—Core Components*
appref	*IBM DCE Version 3.2 for AIX and Solaris: Application Development Reference*
comref	*IBM DCE Version 3.2 for AIX and Solaris: Administration Commands Reference*
dceintro	*IBM DCE Version 3.2 for AIX and Solaris: Introduction to DCE*
dirsrv	*IBM DCE Version 3.2 for AIX and Solaris: Application Development Guide—Directory Services*
hacmp	*IBM DCE Version 3.2 for AIX: High Availability Cluster Multi-Processing Guide for DCE and DFS*
ldaprgy	*IBM DCE Version 3.2 for AIX and Solaris: DCE Security Registry and LDAP Integration Guide*
pdg	*IBM DCE Version 3.2 for AIX and Solaris: Problem Determination Guide*

The following files are contained in the Online Documentation package:

Notes:

1. **en_US** can be substituted with the appropriate locale name.
2. Not all books are translated for each language. Non-translated books will view in English.
3. Individual files are extracted from the tar file at install time (HTML only).

Fileset:

dce.doc.en_US.ascii

/usr/lpp/dcedoc/3270/en_US/booklist
/usr/lpp/dcedoc/3270/en_US/admingd.list3270
/usr/lpp/dcedoc/3270/en_US/admintro.list3270
/usr/lpp/dcedoc/3270/en_US/aixquick.list3270

/usr/lpp/dcedoc/3270/en_US/apgstyle.list3270
/usr/lpp/dcedoc/3270/en_US/appdev.list3270
/usr/lpp/dcedoc/3270/en_US/appref.list3270
/usr/lpp/dcedoc/3270/en_US/comref.list3270
/usr/lpp/dcedoc/3270/en_US/dceintro.list3270
/usr/lpp/dcedoc/3270/en_US/dirsrv.list3270
/usr/lpp/dcedoc/3270/en_US/hacmp.list3270
/usr/lpp/dcedoc/3270/en_US/ldaprgy.list3270
/usr/lpp/dcedoc/3270/en_US/pdg.list3270

Fileset:

dce.doc.en_US.html

/usr/lpp/dcedoc/html/en_US/index.html
/usr/lpp/dcedoc/html/en_US/masthead.gif
/usr/lpp/dcedoc/html/en_US/backgr.jpg
/usr/lpp/dcedoc/html/en_US/ADMINGD/ADMINGD.tar.Z
/usr/lpp/dcedoc/html/en_US/ADMINTRO/ADMINTRO.tar.Z
/usr/lpp/dcedoc/html/en_US/AIXQUICK/AIXQUICK.tar.Z
/usr/lpp/dcedoc/html/en_US/APGSTYLE/APGSTYLE.tar.Z
/usr/lpp/dcedoc/html/en_US/APPDEV/APPDEV.tar.Z
/usr/lpp/dcedoc/html/en_US/APPREF/APPREF.tar.Z
/usr/lpp/dcedoc/html/en_US/COMREF/COMREF.tar.Z
/usr/lpp/dcedoc/html/en_US/DCEINTRO/DCEINTRO.tar.Z
/usr/lpp/dcedoc/html/en_US/DIRSRV/DIRSRV.tar.Z
/usr/lpp/dcedoc/html/en_US/HACMP/HACMP.tar.Z
/usr/lpp/dcedoc/html/en_US/LDAPRGY/LDAPRGY.tar.Z
/usr/lpp/dcedoc/html/en_US/PDG/PDG.tar.Z

Fileset:

dce.doc.en_US.pdf

/usr/lpp/dcedoc/pdf/en_US/booklist
/usr/lpp/dcedoc/pdf/en_US/admingd.pdf
/usr/lpp/dcedoc/pdf/en_US/admintro.pdf
/usr/lpp/dcedoc/pdf/en_US/aixquick.pdf
/usr/lpp/dcedoc/pdf/en_US/apgstyle.pdf
/usr/lpp/dcedoc/pdf/en_US/appdev.pdf
/usr/lpp/dcedoc/pdf/en_US/appref.pdf
/usr/lpp/dcedoc/pdf/en_US/comref.pdf
/usr/lpp/dcedoc/pdf/en_US/dceintro.pdf
/usr/lpp/dcedoc/pdf/en_US/dirsrv.pdf
/usr/lpp/dcedoc/pdf/en_US/hacmp.pdf
/usr/lpp/dcedoc/pdf/en_US/ldaprgy.pdf
/usr/lpp/dcedoc/pdf/en_US/pdg.pdf

Fileset:

dce.doc.rte.ascii

/usr/lpp/dcedoc/bin/asciiview
/usr/lpp/dcedoc/bin/dceman
/usr/lpp/dcedoc/bin/start_dcedoc

Appendix B. DCE Web Secure for AIX Advanced Configuration

These instructions cover advanced configuration for DCE Web Secure. These steps assume that your installation and basic configuration were successful.

Advanced configuration involves manually changing the **magnus.conf** and **obj.conf** files for your Netscape Web server. You should be knowledgeable in the syntax and context of these files before making any changes. Also, save a backup copy of the files should your configuration changes not perform to your expectations. Consult your Netscape documentation or the Netscape Web site for more information on the Netscape configuration files.

To modify the **obj.conf** with the iPlanet Web Server Administration Server GUI after a Web Secure configuration, the user must synchronize the **obj.conf** file before making changes. Click Apply to load the Load Configuration Files that synchronize the **obj.conf** file. This synchronization ensures that the **obj.conf** file remains consistent with the copy maintained in the iPlanet Web Server Administration Server GUI.

Authenticated Path Configuration

You can specify additional paths that need DCE credentials before access is granted by using the **auth-path** parameter on the PathCheck directive. The path specified is evaluated against translated paths resulting from NameTrans directives in the **obj.conf** file.

You can specify auth-path for the following reasons:

- The path contains DCE-enabled CGI programs.
- The path needs authentication with a DCE keytab file rather than through basic authentication, if used in conjunction with the **keyfile** parameter.

Keyfile Configuration

When a DCE administrator wants users to access paths that require DCE credentials, such as to a directory containing DCE-enabled CGIs, and wants the user to operate under a specific DCE userid, the DCE administrator can create a keytab file and specify that it be used for a particular path.

DCE Web Secure allows the use of keytab files by adding the **keyfile**=″keytab-file-path″ and **keyfile-user**=″user-in-keytab-file″ parameters to the dce-restrict PathCheck directive in the Web server's **obj.conf** file.

Summary of Advanced Configuration Syntax

```
PathCheck fn="dce-restrict" <auth-path="path">
 <keyfile="keytab-file-path"
keyfile-user="user-in-keytab-file">
```

The **dce-restrict** PathCheck directive can specify valid combinations of the following optional arguments:

auth-path
: Specifies a path prefix that describes objects in the file system that need DCE credentials before access is granted.

keyfile
: Indicates that authentication on objects that match the associated auth-path should be performed through the specified keytab file. The keyfile must be an existing, valid DCE keytab file. The keyfile parameter must always be used in conjunction with the keyfile-user parameter.

keyfile-user
: Specifies a valid DCE principal in the keyfile keytab file. The keyfile-user parameter must always be used in conjunction with the keyfile parameter.

Examples of Advanced Configuration

The following example shows multiple **dce-restrict** PathCheck directives that might reside in a Web server's **obj.conf** file.

```
PathCheck fn="dce-restrict" auth-path="/mycgidir"
PathCheck fn="dce-restrict" auth-path="/keytest/cgi"
      keyfile="/var/keyfile" keyfile-user="cgi_server_dceid"
```

The first statement is an example of how a CGI application might be configured so that DCE authentication is provided when CGI programs in **/mycgidir** are executed. The second statement is an example of how to specify a keytab file for authentication. For example, the following statements were used to set up the keytab file:

```
$ su (Netscape-server-userid)
$ rgy_edit
rgy_edit> kta -p cgi_server_dceid -pw (random-password) -f /var/keyfile
rgy_edit> exit
```

To test whether the keyfile is set up correctly, these statements were used:

```
$ su (Netscape-server-userid)
$ dce_login cgi_server_dceid -k /var/keyfile
```

Appendix C. Dummy Filesets

The following is a list of ″dummy″ filesets. These filesets existed in previous DCE releases. They have either been renamed or merged with other filesets in the IBM DCE V3.2 for AIX release:

```
dce.client.core.rte
dce.client.core.rte.rpc
dce.pthreads.rte
dce.web.admin.rte
dce.web.secure.rte
dce.compat.cds.smit
dce.compat.client.core.smit
dce.compat.security.smit
dce.compat.sysmgmt.ems.smit
dce.compat.sysmgmt.snmpagt.smit
dce.compat.web.admin.smit
dce.msg.en_US.client.core.rte
dce.msg.en_US.compat.cds.smit
dce.msg.en_US.compat.client.core.smit
dce.msg.en_US.compat.security.smit
dce.msg.en_US.compat.sysmgmt.ems.smit
dce.msg.en_US.compat.sysmgmt.snmpagt.smit
dce.msg.en_US.pthreads.rte
dce.msg.en_US.web.admin.rte
dce.msg.en_US.web.secure.rte
dce.msg.es_ES.client.core.rte
dce.msg.es_ES.compat.cds.smit
dce.msg.es_ES.compat.client.core.smit
dce.msg.es_ES.compat.security.smit
dce.msg.es_ES.compat.sysmgmt.ems.smit
dce.msg.es_ES.compat.sysmgmt.snmpagt.smit
dce.msg.es_ES.pthreads.rte
dce.msg.es_ES.web.admin.rte
dce.msg.es_ES.web.secure.rte
dce.msg.Es_ES.client.core.rte
dce.msg.Es_ES.compat.dcs.smit
dce.msg.Es_ES.compat.client.core.smit
dce.msg.Es_ES.compat.security.smit
dce.msg.Es_ES.compat.sysmgmt.ems.smit
dce.msg.Es_ES.compat.sysmgmt.snmpagt.smit
dce.msg.Es_ES.pthreads.rte
dce.msg.Es_ES.web.admin.rte
dce.msg.Es_ES.web.secure.rte
dce.msg.ja_JP.client.core.rte
dce.msg.ja_JP.compat.cds.smit
dce.msg.ja_JP.compat.client.core.smit
dce.msg.ja_JP.compat.security.smit
dce.msg.ja_JP.compat.sysmgmt.ems.smit
dce.msg.ja_JP.compat.sysmgmt.snmpagt.smit
dce.msg.ja_JP.pthreads.rte
```

```
dce.msg.ja_JP.web.admin.rte
dce.msg.ja_JP.web.secure.rte
dce.msg.Ja_JP.client.core.rte
dce.msg.Ja_JP.compat.cds.smit
dce.msg.Ja_JP.compat.client.core.smit
dce.msg.Ja_JP.compat.security.smit
dce.msg.Ja_JP.compat.sysmgmt.ems.smit
dce.msg.Ja_JP.compat.sysmgmt.snmpagt.smit
dce.msg.Ja_JP.pthreads.rte
dce.msg.Ja_JP.web.admin.rte
dce.msg.Ja_JP.web.secure.rte
dce.msg.ko_KR.client.rte
dce.msg.ko_KR.compat.cds.smit
dce.msg.ko_KR.compat.client.core.smit
dce.msg.ko_KR.compat.security.smit
dce.msg.ko_KR.compat.sysmgmt.ems.smit
dce.msg.ko_KR.compat.sysmgmt.snmpagt.smit
dce.msg.ko_KR.pthreads.rte
dce.msg.ko_KR.web.admin.rte
dce.msg.ko_KR.web.secure.rte
dce.msg.zh_TW.client.core.rte
dce.msg.zh_TW.compat.cds.smit
dce.msg.zh_TW.compat.client.core.smit
dce.msg.zh_TW.compat.security.smit
dce.msg.zh_TW.compat.sysmgmt.ems.smit
dce.msg.zh_TW.compat.sysmgmt.snmpagt.smit
dce.msg.zh_TW.pthreads.rte
dce.msg.zh_TW.web.admin.rte
dce.msg.zh_TW.web.secure.rte
dce.msg.Zh_TW.client.core.rte
dce.msg.Zh_TW.compat.cds.smit
dce.msg.Zh_TW.compat.client.core.smit
dce.msg.Zh_TW.compat.securtity.smit
dce.msg.Zh_TW.compat.sysmgmt.ems.smit
dce.msg.Zh_TW.compat.sysmgmt.snmpagt.smit
dce.msg.Zh_TW.pthreads.rte
dce.msg.Zh_TW.web.admin.rte
dce.msg.Zh_TW.web.secure.rte
```

Appendix D. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make

improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *2001.* All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

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

AIX
DFS
IBM
RISC System/6000
SecureWay

UNIX is a registered trademark of X/Open Company Limited.

Windows NT is a registered trademark of Microsoft Corporation.

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

Index

Special Characters

/opt/dcelocal subtree 47
/var/dce 49

A

access control 34
ACLs, container 18
added commands
  CDS
    cdsdel 12
    cdsli 12
  configuration
    chpesite 11
    config.dce 11
    mkreg.dce 12
    rmreg.dce 12
    show.cfg 12
    unconfig.dce 12
  RPC
    rpcprotseqs 12
    rpcresolve 12
  security
    rmxcred 12
added services
  documentation 124
additional file systems to create 49
admin client configuration 86
administration programs 44
  cdscp 45
  cdsdel 45
  cdsli 45
  dcecp 44, 45
  DTS 45
  group_override 45
  passwd_export 45
  passwd_import 44
  passwd_override 45
  registry 45
  rmxcred 45
  rpc 44
AES/Distributed Computing - Directory Services 13
AES/Distributed Computing - Remote Procedure Call 13
AES/Distributed Computing - Security 13
AES/Distributed Computing - Threads 13
AES/Distributed Computing - Time Services 13
AIX programs
  DCE Base Services for AIX 5
  DCE Cell Directory Server for AIX 9
  DCE Security Services for AIX 9
  Privacy Level Protection 9
AIX standard accounts 26
application development 46
audit service
  about 7
  AIX programs 7
  audit application programming interfaces 7
  audit daemon 7
  audit management interfaces 7
  programs, AIX 7

B

bibliography 124
  DCE Publications
    IBM 124
  help files 124
  online books 123

C

cds-admin group 34
CDS clerk 37
CDS client
  configuring 93
  master security server 93
CDS Preferencing 6
cdsadv 37, 42
cdsclerk 37
cdscp 37, 45
cdsd 42
cdsdel 45
cdsli 45
cell
  definition 3
  planning 24
cell configuration
  audit servers 104
  configuring DCE clients 86
  configuring GDA 101
  configuring secondary security servers 99
  configuring security replica servers 99
cell configuration *(continued)*
  DCE 3.2 for AIX EMS server 102
  DCE 3.2 for AIX security integration 103
  DCE 3.2 for AIX SNMP server 103
  Identity Mapping Server 108
  introduction 94
  Name Service Interface Daemon (NSID) 107
  Password Strength Servers 105
  secondary CDS servers 97
cell namespace
  boundaries 29
  entries 30
  stability 30
cell-relative names 28
chpesite 75
cleanup.dce 75
client programs
  CDS 37
  DTS 37
  RPC 36
config.dce 75
configuration 51
  chpesite 75
  cleanup.dce 75
  clock skew 75
  config.dce 75
  further cell configuration 94
  initial cell configuration 73
  kerberos.dce 11, 75
  migrate.dce 11, 76
  minimum requirements 73
  mkdceweb 11, 76
  mkreg.dce 76
  overview 73
  rmdceweb 12, 76
  rmreg.dce 76
  show.cfg 76
  start.dce 76
  stop.dce 76
  unconfig.dce 76
  using SMIT 75
configuring
  audit servers 104
  CDS client 93

configuring *(continued)*
DCE 3.2 for AIX EMS server 102
DCE 3.2 for AIX security integration 103
DCE 3.2 for AIX SNMP server 103
DCE clients 86
DCE Web Secure 109
DTS client 97
DTS servers 94
GDA 101
Identity Mapping Server 108
initial CDS server 83
initial cell 80
master security server 81, 93
Name Service Interface Daemon (NSID) 107
Password Strength Servers 105
secondary CDS servers 97
secondary security servers 99
security replica servers 99
conformance to standards 13
container ACLs 18
control program 44
create, file systems 49

D

daemons
cdsd 42
dtsd 37, 43
gdad 42, 43
secd 40
DCE
description 3
DCE ACLs
differences between DCE and AIX 15
DCE aliases 18
DCE Audit Services for AIX 7
DCE compatibility with AIX
application core files 27
debugging 15
man command unsupported 13
security 14
DCE for Application Developers (dcetools) 8
dce_hostname 80
DCE Online Documentation 8
DCE Security Services for AIX 9
DCE system management 7
DCE Web Administration 8
Event Management Service (EMS) 7
Simple Network Management Protocol (SNMP) 8
DCE Threads Compatibility Library for AIX 6
DCE Web Administration 46
DCE Web Secure 46
dcecp 44, 45
dfs-admin group 34
disk space required (MB) 24
DNS global names 27
DTS
configuring servers 94
planning 43
dts-admin group 34
dtsd 37, 43

E

easy installation program 58

F

file location
/opt/dcelocal 47
UNIX subdirectories 47
files
to create after installation 49
full client configuration 89

G

GDA
planning 42
processes 43
gda_child 43
gdad 42
global names
DCE cell name 28
obtaining 29
global planning 24
group_override 45
groups 34

I

idl compiler 46
information
ordering publications 124
initial CDS server
configuring 83
initial cell configuration 80
CDS server 83
DTS servers 94
master security server 81, 93
installation 51
disk space required (MB) 24
program, easy 58
requisite software 54
stopping processes 58
installp 60

K

Kerberos 13
kerberos.dce 11, 75

L

limitations 18
container ACLs 18
DCE aliases 18
primary names 19
renaming principals, groups, and orgs 19
sec_admin -s 18
using dcecp catalog commands to search multiple subtrees 19
local client configuration 87

M

man command unsupported 13
master security server
CDS client 93
configuring 81
migrate.dce 11, 76
migrating
before 61
permissions 61
mkdceweb 11, 76
mkreg.dce 76
multithreaded applications 15
multithreaded programming environment 6

N

names
cell 27, 29
cell-relative 28
namespace
cell 33
clearinghouse 31
definition 31
entry types 32
introduction 29
planning 24
replication 33
security 32
NTP 13

O

O'Reilly & Associates books 124
Online Documentation 8

P

packaging
AIX programs 6
DCE Threads for AIX Compatibility Library 6
programs, AIX 6
passwd_export 45
passwd_import 44
passwd_override 45
password strength server 7

POSIX 13
primary names, storing in LDAP 19
Privacy Level Protection 9
profiles, CDS namespace 32
programs, AIX
  DCE Base Services for AIX 5
  DCE Cell Directory Server for AIX 9
  DCE Security Services for AIX 9
  Privacy Level Protection 9
publications 123

Q

questions for planning 24

R

registry 45
renaming principals, groups, and orgs 19
requisite software 54
RFC 1006 13
RFC 1129 13
rmdceweb 12, 76
rmreg.dce 76
rmxcred 45
rpcprotseqs 12
rpcresolve 12

S

searching multiple subtrees 19
sec-admin group 34
secd 40
security 34
security service
  password strength server 7
server processes
  CDS 42
  DTS 43
  security 40
show.cfg 76
SMIT 7
smitty 7
split configuration of clients
  admin 86
  full 89
  local 87
standards conformance 13
start.dce 76
start.dce all 119
start.dce core 119
starting DCE
  using command line 119
  using SMIT 120
stop.dce 76, 121
stop.dce all 121
stop.dce core 121
stopping DCE 121
stopping processes for installation 58

T

technology components
  Directory Service 9
  Distributed Time Service 6
  multithreaded programming environment 6
  RPC 6
  security client
    extended registry attributes 7
    GSSAPI extensions 6
  Security client 6
    AIX security integration 6
  XDS/XOM 8
TPO-to-TCP 13

U

unconfig.dce 76
unconfiguring
  before 113
  introduction 112
  split unconfiguration of clients 114
  steps 115
UNIX directories 47
unsupported OSF features
  commands
    configuration 17
    dce_config 17
    dtss-graph 17
    sec_salvage_db 17
    security 17
    user commands 17

W

warnings
  two machines with same dce_hostname 80
  unconfiguring secondary CDS server 113

Part Number: CT6AHNA

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

(1P) P/N: CT6AHNA