



AS/400e series

# IBM Distributed Computing Environment Version 2.2, Base Services for AS/400: Getting Started

SC41-5601-00





AS/400e series

# IBM Distributed Computing Environment Version 2.2, Base Services for AS/400: Getting Started

**Note**

Before using this document, read the information in “Appendix F. Notices” on page 73.

**First Edition (September 1998)**

This edition applies to Version 4 Release 3 of the IBM Distributed Computing Environment Version 2.2, Base Services for AS/400, (5769-DC1) and to all subsequent releases and modifications until otherwise indicated in new editions. This edition applies only to reduced instruction set computer (RISC) systems.

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

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

---

# Contents

<b>Welcome to Distributed Computing Environment Base Services for AS/400</b>	<b>v</b>
Who Should Read This Book . . . . .	vi
Conventions and Terminology Used in This Book . . . . .	vi
AS/400 Operations Navigator . . . . .	vii
Installing Operations Navigator subcomponents . . . . .	viii
Accessing AS/400 Operations Navigator . . . . .	ix
Prerequisite and related information . . . . .	ix
How to send your comments. . . . .	ix
<b>Chapter 1. Overview of DCE Version 2.2, Base Services for AS/400</b>	<b>1</b>
What Is DCE?. . . . .	1
The DCE for AS/400 Product . . . . .	2
<b>Chapter 2. System Requirements.</b>	<b>3</b>
<b>Chapter 3. Product Components</b>	<b>5</b>
Determining Requirements for DCE Client Machines . . . . .	5
Remote Procedure Call (RPC) . . . . .	5
CDS Client . . . . .	5
Security Client . . . . .	5
Distributed Time Service (DTS) . . . . .	6
<b>Chapter 4. Setting Up DCE for AS/400</b>	<b>9</b>
<b>Chapter 5. Configuring DCE for AS/400</b>	<b>11</b>
DCE Cells . . . . .	11
Defining a Hostname . . . . .	11
Joining a DCE Cell . . . . .	11
Running DCE Configuration . . . . .	12
Configuring DCE on Your System . . . . .	14
Configuring a Full Client . . . . .	15
Optional QSHELL Command . . . . .	18
Configuring a Local Client . . . . .	19
Optional QSHELL Command . . . . .	20
Changing Your System Configuration . . . . .	21
Unconfiguring Your System . . . . .	22
Full Client Unconfiguration . . . . .	22
Optional QSHELL Command . . . . .	23
Local Client Unconfiguration. . . . .	24
Optional QSHELL Command . . . . .	24
Checking the Configuration Status . . . . .	24
Optional QSHELL Command . . . . .	26
Viewing the Log File . . . . .	26
Viewing and Correcting Errors . . . . .	28
Starting, Stopping, and Cleaning Up DCE Operations Navigator . . . . .	29
QSHELL Command Line . . . . .	31
CL Command Line . . . . .	31
<b>Chapter 6. DCE Features and Functions</b>	<b>33</b>
Supported Functions in this Release . . . . .	33
Supported DCE and DCECP Commands	34
<b>Chapter 7. Serviceability and Internationalization.</b>	<b>37</b>
XPG4 Internationalization Considerations	37
XPG4 Versus AS/400-Proprietary Internationalization . . . . .	37
XPG4 Locale and Message Catalogs: NLSPATH Environment Variable . . . . .	37
Network CCSID for DCE Communication	39
Summary of NLS Setup Considerations and Programming Considerations . . . . .	42
Setting the XPG4/POSIX Locale. . . . .	42
Sample Messaging Program . . . . .	42
Network CCSID . . . . .	42
NLSPATH . . . . .	43
<b>Appendix A. Helpful Tips</b>	<b>45</b>
Threads . . . . .	45
Running Multithreaded AS/400 DCE Applications . . . . .	45
The DCE dce_free Routine . . . . .	46
dce_free. . . . .	46
Remote Procedure Call (RPC) Memory Management . . . . .	47
Using rpc_ss_allocate and rpc_ss_free in Manager Code . . . . .	48
Using rpc_ss_allocate and rpc_ss_free in Client Code . . . . .	48
Using Your Own Allocation and Free Routines . . . . .	49
IDL Encoding Services Memory Management . . . . .	50

Advanced Memory Management Support . . . . .	50	Required Compiler Flags . . . . .	64
Memory Management for Pointed-to Nodes . . . . .	52	Information about the VisualAge Compiler . . . . .	64
<b>Appendix B. Documentation Provided with this Release . . . . .</b>	<b>55</b>	<b>Appendix D. Support Services. . . . .</b>	<b>67</b>
Viewing the HTML Files . . . . .	57	IBM DCE Base Services for AS/400 . . . . .	67
Using the PDF Files . . . . .	57	<b>Appendix E. Restrictions and Known Problems . . . . .</b>	<b>69</b>
<b>Appendix C. DCE Base Services for AS/400 Application Development Kit . . . . .</b>	<b>59</b>	National Language Support (NLS) Restrictions. . . . .	71
The ADK and DCE for Windows NT . . . . .	59	Message Handling Restrictions and Serviceability Restrictions . . . . .	72
Installing the ADK . . . . .	60	<b>Appendix F. Notices . . . . .</b>	<b>73</b>
The Include Path. . . . .	61	Trademarks . . . . .	75
IDL Compiler . . . . .	61	<b>Index . . . . .</b>	<b>77</b>
Stub Auxiliary Files . . . . .	62	<b>Readers' Comments — We'd Like to Hear from You . . . . .</b>	<b>79</b>
SAMS . . . . .	62		
Example Programs . . . . .	63		
Generating Universal Unique Identifiers (UUID) . . . . .	64		

---

## Welcome to Distributed Computing Environment Base Services for AS/400

This book describes the IBM Distributed Computing Environment Version 2.2, Base Services for AS/400 (DCE for AS/400) client product, and explains how to plan for, configure, and use the product.

“Chapter 1. Overview of DCE Version 2.2, Base Services for AS/400” on page 1 gives a brief overview of DCE for AS/400.

“Chapter 2. System Requirements” on page 3 lists the products that are compatible with DCE for AS/400.

“Chapter 3. Product Components” on page 5 describes configurations for DCE client machines.

“Chapter 4. Setting Up DCE for AS/400” on page 9 explains how to install DCE for AS/400.

“Chapter 5. Configuring DCE for AS/400” on page 11 explains how to configure DCE for AS/400.

“Chapter 6. DCE Features and Functions” on page 33 discusses which functions and features from the DCE 2.2 for AIX product are supported in this release.

“Chapter 7. Serviceability and Internationalization” on page 37 discusses XPG4 and CCSID considerations.

“Appendix A. Helpful Tips” on page 45 provides information that will help resolve some of the more common problems you might encounter such as freeing allocated memory.

“Appendix B. Documentation Provided with this Release” on page 55 provides information about the additional documentation that is provided for this release.

“Appendix C. DCE Base Services for AS/400 Application Development Kit” on page 59 discusses how to install and use the DCE for AS/400 Application Developers Toolkit.

“Appendix D. Support Services” on page 67 discusses the types of support services that are provided with the IBM DCE Base Services for AS/400 and with the IBM DCE Base Services for AS/400 Application Development Kit (ADK).

“Appendix E. Restrictions and Known Problems” on page 69 discusses restrictions and known problems in this release.

---

## Who Should Read This Book

This publication is intended for use by AS/400 system administrators who have a basic understanding of the Distributed Computing Environment.

---

## Conventions and Terminology Used in This Book

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. **Bold** is also used for emphasis.

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

**Constant width**

Examples and information that the system displays appear in constant width type style.

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

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

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

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

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

The AS/400 displays in this book could be shown as they are presented through Graphical Access for AS/400, which is part of Client Access on the personal computer. The example displays in this book could also be shown without Graphical Access for AS/400 available. Figure 1 on page vii shows both types of displays.



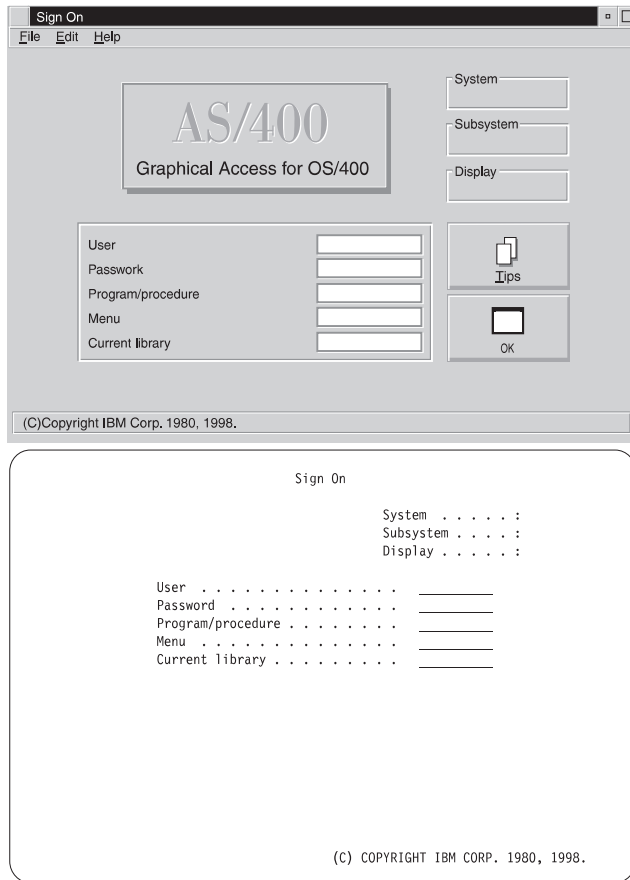


Figure 1. Types of AS/400 Displays

## AS/400 Operations Navigator

AS/400 Operations Navigator is a powerful graphical interface for Windows 95/NT clients. With AS/400 Operations Navigator, you can use your Windows 95/NT skills to manage and administer your AS/400 systems.

- You can work with basic operations (messages, printer output, and printers), job management, system configuration, network administration, security, users and groups, database administration, file systems, and multimedia.
- You can schedule regular system backups, work with Interprocess Communication through application development, and manage multiple AS/400 systems through a central system by using Management Central. You can also customize the amount of Operations Navigator function that a user or user group can use through application administration.

- You can create a shortcut to any item in the explorer view of Operations Navigator. For example, you can create a shortcut either to **Basic Operations** or to the items that are listed under **Basic Operations (Messages, Printer Output, and Printers)**. You can even create a shortcut to an individual printer or use a shortcut as a fast way to open the item.

Figure 2 shows an example of the Operations Navigator display:

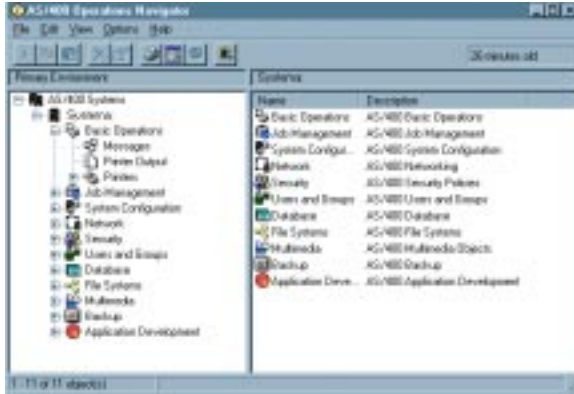


Figure 2. AS/400 Operations Navigator Display

IBM recommends that you use this new interface. It has online help to guide you. While we develop this interface, you will still need to use either of the following to do some of your tasks:

- Graphical Access (which provides a graphical interface to AS/400 screens). Graphical Access is part of the base Client Access.
- A traditional emulator such as PC5250.

## Installing Operations Navigator subcomponents

AS/400 Operations Navigator is packaged as separately installable subcomponents. If you are upgrading from a previous release of AS/400 Operations Navigator, only those subcomponents that correspond to the function that is contained in the previous release will be installed. If you are installing for the first time and you use the **Typical** or **Minimum** installation options, the following options are installed by default:

- Operations Navigator base support
- Basic operations (messages, printer output, and printers)

To install additional AS/400 Operations Navigator subcomponents, either use the **Custom** installation option or use **selective setup** to add subcomponents after Operations Navigator has been installed:

1. Display the list of currently installed subcomponents in the **Component Selection** window of **Custom** installation or selective setup.
2. Select AS/400 Operations Navigator and click **Details**.
3. Select any additional subcomponents that you want to install and continue with **Custom** installation or selective setup.

**Note:** To use AS/400 Operations Navigator, you must have Client Access installed on your Windows 95/NT PC and have an AS/400 connection from that PC. For help in connecting your Windows 95/NT PC to your AS/400 system, consult *Client Access for Windows 95/NT - Setup, SC41-3512*.

## Accessing AS/400 Operations Navigator

To access Operations Navigator after you install Client Access and create an AS/400 connection, do the following:

1. Double-click the **Client Access** folder on your desktop.
2. Double-click the **Operations Navigator** icon to open Operations Navigator. You can also drag the icon to your desktop for even quicker access.

---

## Prerequisite and related information

Use the AS/400 Information Center as a starting point for your AS/400 information needs. It is available in either of the following ways:

- The Internet at this uniform resource locator (URL) address:  
<http://publib.boulder.ibm.com/html/as400/infocenter.html>
- On CD-ROM: *AS/400e series Information Center, SK3T-2027*.

The AS/400 Information Center contains browsable information on important topics such as Java, program temporary fixes (PTFs), and Internet security. It also contains hypertext links to related topics, including Internet links to Web sites such as the AS/400 Technical Studio, the AS/400 Softcopy Library, and the AS/400 home page.

For a list of related publications, see the “Appendix B. Documentation Provided with this Release” on page 55.

---

## How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other AS/400 documentation, fill out the readers' comment form at the back of this book.

- If you prefer to send comments by mail, use the readers' comment form with the address that is printed on the back. If you are mailing a readers' comment form from a country other than the United States, you can give the form to the local IBM branch office or IBM representative for postage-paid mailing.
- If you prefer to send comments by FAX, use either of the following numbers:
  - United States and Canada: 1-800-937-3430
  - Other countries: 1-507-253-5192
- If you prefer to send comments electronically, use this network ID:
  - IBMMAIL, to IBMMAIL(USIB56RZ)
  - RCHCLERK@us.ibm.com

Be sure to include the following:

- The name of the book.
- The publication number of the book.
- The page number or topic to which your comment applies.

---

# Chapter 1. Overview of DCE Version 2.2, Base Services for AS/400

IBM Distributed Computing Environment Version 2.2, Base Services for AS/400 (DCE for AS/400) is a member of the IBM Server Series family of products. This DCE for AS/400 release is based on the Distributed Computing Environment (DCE) 2.2 for AIX version of the Open Software Foundation (OSF) DCE 1.2.2 product.

---

## What Is DCE?

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

An application that is 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 also hides the complexity of the network environment.

Distributed computing services, as put into effect 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 layer and the distributed application program layer. DCE provides a variety of common services that are needed for development of distributed applications, such as 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.

The term *cell* refers to a group of DCE machines that work together and that are administered as a unit. For example, imagine an organization that is comprised of several departments, each residing 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.

You can configure a DCE cell in many ways, depending on your requirements. A cell consists of a network that connects 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** contain special software that enables them to provide one or more of the DCE services.

DCE for AS/400 is a layer between the AS/400 operating system and network services layer, and a distributed application layer. It 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 for AS/400 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.

---

## The DCE for AS/400 Product

The DCE for AS/400 product includes the following components:

- Remote Procedure Call (RPC)
- Cell Directory Service (CDS) client
- Security client
- Distributed Time Service (DTS)
- Application servers

See "Chapter 3. Product Components" on page 5 for more detailed information about the individual components.

---

## Chapter 2. System Requirements

The system requirements for this product are an AS/400 system that runs following:

- Version 4 Release 3 of 5769SS1

**Note:** This product will not work on prior versions of the AS/400 operating system.

In addition to the base AS/400 product, you need these optionally installable parts of 5769SS1 :

- Option 12 — Host Servers
- Option 13 — System Openness Includes (This is needed only if you want to use the Application Developer's Tool Kit.)
- Option 30 — QShell Interpreter
- 5716CX5 — VA C++ for Windows 95/NT (This is needed only if you want to use the Application Developer's Tool Kit.)
- 5763XD1 — Client Access/400 Optimized for Windows
- 5769TC1 — Transmission Control Protocol/Internet Protocol (TCP/IP)
- 5769XW1 — Client Access/400 Windows Family Base

The DCE cell requires DCE CDS and DCE Security servers on another platform (such as, AIX or Windows NT). You must also know the cell name of an existing cell so that you can configure your client into that cell. See the documentation that came with your operating system for more information.

For configuration you must have a Windows 95 or a Windows NT workstation that has Client Access. The Operations Navigator component of Client Access must have Base Services and Network subcomponents installed.

AS/400 DCE clients interoperate with CDS servers and Security servers that run on the following platforms:

- IBM DCE for AIX v2.2
- IBM DCE for AIX v2.1.0
- Digital DCE for NT v1.1c
- IBM DCE for NT v2.0
- Gradient DCE for NT vX.x
- DCE for Solaris 2.5

In addition, DCE for AS/400 application servers support clients that run on the following list of products. Similarly, AS/400 DCE clients support servers that run on the following products:

- IBM DCE for AIX v2.2
- IBM DCE for AIX v2.1.0
- AS/400 DCE 1.03c
- Digital DCE for NT 1.1c
- IBM DCE for NT 2.0
- IBM DCE for Windows 95 v2.0
- IBM DCE for MVS v5.1
- Gradient DCE for NT vX.x
- DCE for Solaris 2.5 (latest Transarc version)
- OS/2 DCE v2.1.0



---

## Chapter 3. Product Components

This section describes configurations for DCE client 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.

---

### 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 daemons that run on a DCE client machine.

#### Remote Procedure Call (RPC)

**dced** must run on any machine with an RPC server process. The process must export an interface with dynamic bindings. **dced** registers binding information.

**dced** 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** can run on a machine at a time, because **dced** uses a well-known port.

#### CDS Client

The DCE client runs the following CDS processes:

- The CDS Advertiser (**cadsadv**) allows applications to access and communicate with a CDS server, starts any needed CDS clerks, and creates the cache shared by local CDS clerks.
- The CDS clerk (**cadsclerk**) is an interface between CDS client applications and CDS servers.

#### Security Client

The security client maintains the local machine principal identity by periodically refreshing the ticket-granting ticket for the machine principal. This assures that the user, or any daemon who inherits the machine identity,

has valid DCE credentials. The security client also exports and puts into effect a variety of interfaces, that includes password and group override support, certification of the security server, and pre-authentication support.

## Distributed Time Service (DTS)

DTS services include the Distributed Time Service (DTS) daemon (**dttd**) and the **dts\_device\_name\_provider**.

### DTS Daemon (dttd)

You can set the **dttd** daemon as a client or a server. As a client, **dttd** synchronizes the local clock. As a server, **dttd** synchronizes with other DTS servers, in addition to synchronizing the local clock.

### **dts\_device\_name\_provider**

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. You need at least three DTS servers 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 that are 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 that are 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 will prevent a warning message from being logged every time the server attempts to synchronize.

There are many network configuration decisions that affect DTS planning. The *DCE Administration Guide - Core Components* contains details about the total

DTS planning process. This process includes configuration planning for local area networks (LANs), extended LANs, and wide area networks (WANs). The *DCE Administration Guide - Core Components* also contains an explanation of the criteria you need to use when selecting a time source for your network to use.



---

## Chapter 4. Setting Up DCE for AS/400

To install DCE for AS/400, see the *AS/400 Software Installation* book, SC41-. This book explains how to install AS/400 licensed products.

After the product has been installed, as a privileged user profile (for example, QSECOFR), you must enable and set an initial password for the QDCE user profile. Enable and set the initial password by issuing the following CL command on the AS/400 system:

```
CHGUSRPRF USRPRF(QDCE) STATUS(*ENABLED) PASSWORD(XXXX)
```

There are multiple ways to set the locale, including letting it default to the POSIX/C or the AS/400 system setting. DCE supports both of these settings. DCE does not require that the locale be set by using the user profile.

If you want to set the locale through the QDCE user profile, issue the following command:

```
CHGUSRPRF USRPRF(QDCE) LOCALE('/QSYS.LIB/XX_XX.LOCALE')
```

where XX\_XX is the Language\_Country code (for example EN\_US).

You **must** use the QDCE user profile to configure, start, stop, and clean up DCE from the Operations Navigator.

**Note:** You must also use the QDCE user profile to run the two security APIs that are privileged operations. These two APIs are **sec\_login\_certify\_identity** and **sec\_login\_valid\_and\_cert\_ident**. See the *IBM DCE for AIX, Version 2.2: Application Development Reference* for more information about these APIs.

If a DCE application created with an existing DCE licensed product (5798-TBF) is to be used with the new version, it must first be migrated to the new release of DCE. For more information on applications, see "Appendix C. DCE Base Services for AS/400 Application Development Kit" on page 59.

Because there is no migration for client configuration provided for an AS/400 that is configured into a DCE cell with an existing DCE licensed product, the client must first be **admin** unconfigured from the cell.

The **admin** unconfiguration removes information about the client from the namespace and the security registry. The cell administrator must run the **unconfig.dce** command from a machine within the existing cell. The administrator cannot run the command from the client machine that is being

unconfigured. The cell administrator must be logged on as QDCE on the machine from which the administrator is issuing the command. However, the cell administrator does not need QDCE authority on the client that is being unconfigured. After the **admin** unconfiguration fully removes the host from the cell. You can then reconfigure the client back into the cell and use the new version of DCE.

---

## Chapter 5. Configuring DCE for AS/400

Use the Operations Navigator to perform DCE configuration. This tool provides an easy-to-use graphical user interface for configuring DCE for AS/400 client services from your Windows 95 or Windows NT system.

**Note:** For the necessary Operations Navigator prerequisites, see “Chapter 2. System Requirements” on page 3.

DCE configuration can configure these components so that your AS/400 system can function as a DCE for AS/400 client system. Many of the text entry fields that you encounter during configuration have default values associated with them. These default values are based on your existing configuration if you have one. Otherwise, DCE configuration provides values that are appropriate for the most common DCE configurations. You must be logged in using the user profile QDCE to perform a DCE for AS/400 configuration or to make changes to an existing configuration.

---

### DCE Cells

DCE for AS/400 configuration on Windows 95 or Windows NT lets you configure your system as a DCE client. You can create a configuration that will join an existing cell.

#### Defining a Hostname

All DCE configurations require you to provide a name for your system that is unique within your DCE cell. In most cases, you should use the default hostname. The default hostname is the fully qualified Internet hostname.

To perform DCE full client configurations, you must provide DCE configuration with a principal name and password. For most client system configurations, the principal name you provide must have privileges to perform cell administration operations. For local client configurations, however, you do not need the principal name and the password.

#### Joining a DCE Cell

You need the following information to configure your AS/400 client in an existing DCE cell:

- Cell name

- Host identification of the Master Security server (if a CDS server for the cell or a proxy for a CDS server is not in the broadcast range)
- Host identification of any CDS server in the cell (if a CDS server for the cell or a proxy for a CDS server is not in the broadcast range)
- Security principal name and password that are authorized to perform cell administration operations. (You do not need the security principal name and password if you are performing a local client configuration.)

**Note:**

The Host identification can be either the TCP/IP hostname or the IP address. For any hostname that is not currently defined in the TCP/IP hosts database, the IP address must be provided for the host identification.

---

## Running DCE Configuration

Before starting DCE configuration ensure that you are connected to the AS/400 using the QDCE user profile. Also ensure that the Network subcomponent of the Operations Navigator is installed on your Windows 95 or Windows NT system.

To start DCE configuration from the **AS/400 Operations Navigator** folder:



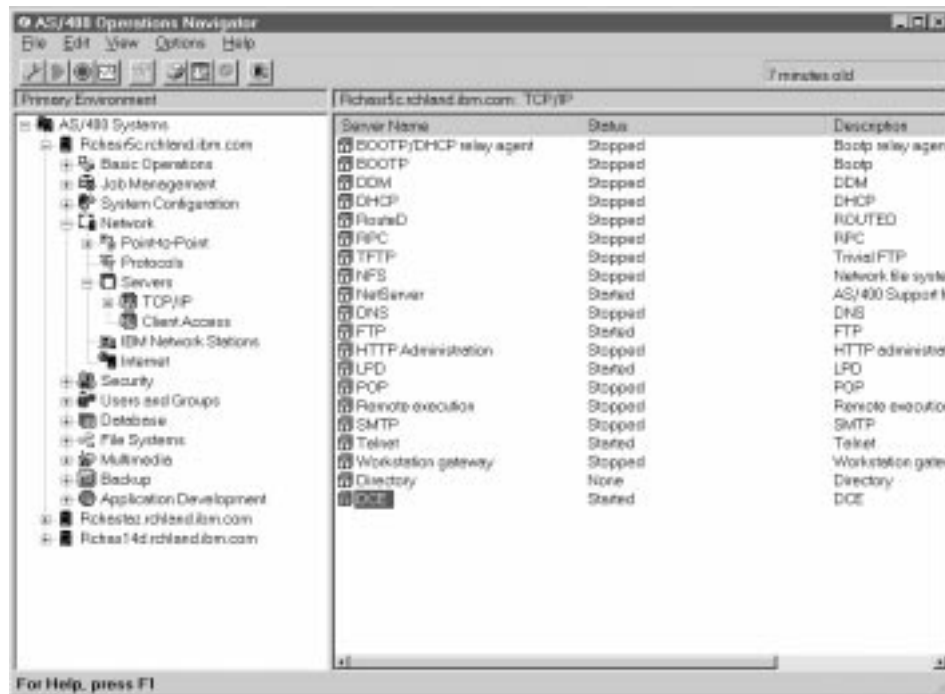


Figure 3. The Operations Navigator panel

1. Double-click on the desired machine.
2. Double-click on **Network**.
3. Double-click on **Servers**.
4. Double-click on **TCP/IP**.
5. Double-click on **DCE**.

The **DCE Configuration** panel is displayed.



Figure 4. The DCE Configuration panel

**Note:** To exit DCE configuration, click on the **File** menu and then click on **Exit**. There is no need to explicitly save any data before exiting. Your configuration is saved automatically.

---

## Configuring DCE on Your System

DCE configuration allows you to choose between two types of client configurations:

- *Full DCE client configuration*, which requires you to supply additional information. This option requires the cell administrator's principal account name and password.
- *Local DCE client configuration*, which uses configuration information previously supplied and stored by a cell administrator while performing an **admin** configuration from a server within the cell.

**Note:** An **admin** configuration from the server must be performed before you can perform a local DCE client configuration. 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. The

administrator cannot run the command from the new client machine. See the DCE documentation for your server for more information on **admin** configuration.

Both options automatically configure your system as a CDS client and a Security client. If you click the Cancel button at any time during configuration, the configuration ends. Configuration will take effect only when you run through all screens and click on **Finish** on the last screen. When you click on **Finish**, DCE configuration configures the system according to the options you selected.

---

## Configuring a Full Client

The **full client configuration** option allows you to configure your system as a DCE client.

When the full client configuration is complete, DCE configuration remembers the options you selected and the information you entered.

Full client configuration requires knowledge of the cell administrator's password. To perform a full client configuration, from the **DCE Configuration** panel, click on the **Configuration** menu, click on **Create**, and then click on **Full Client**. The **Full DCE Client Configuration** panel is displayed.

The screenshot shows a window titled "Full DCE Client Configuration - xyz host". It contains several input fields and a checkbox:

- Cell name:** A single-line text input field.
- Local host:** A container box containing:
  - DCE host name:** A single-line text input field.
  - Host identification:** A single-line text input field.
- Servers:** A container box containing:
  - Security Master server host identification:** A single-line text input field.
  - CDS server host identification:** A single-line text input field.
- Cell administrator:** A container box containing:
  - Principal account name:** A single-line text input field.
  - Principal password:** A single-line text input field.
- Start when TCP/IP is started:** A checkbox.
- Navigation buttons:** Four buttons at the bottom: "< Back", "Next >", "Cancel", and "Help".

Figure 5. The Full DCE Client Configuration panel

The **Cell name**, and the **Principal password** fields are required.

**Note:** If the principal account name is not **cell\_admin** you will need to specify the principal account name in the **Principal account name** field.

DCE for AS/400 provides an option that automatically starts DCE services during TCP/IP startup. When enabled, this feature adds the DCE Auto-Start Service to the list of services that start automatically as part of the startup procedure. If you select this option, you do not have to remember to restart DCE whenever TCP/IP is restarted on your AS/400 system. To start DCE automatically, select **Start when TCP/IP is started**.

After you have completed the required fields and any optional fields that have not been pre-filled by your system, verify that the information is correct. Click on **Next**. The next **Full DCE Client Configuration** panel is displayed.

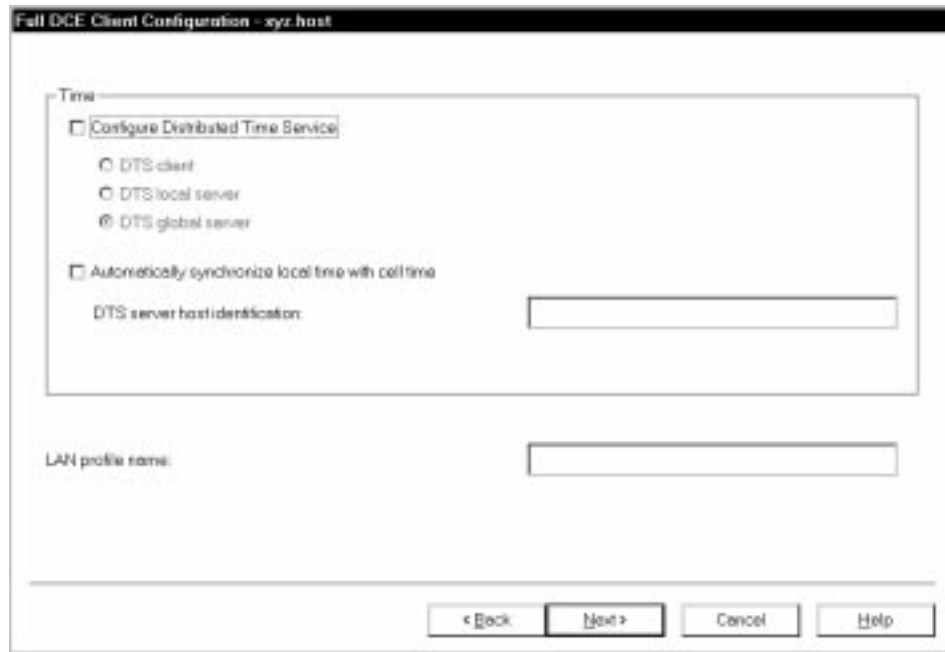


Figure 6. The next Full DCE Client Configuration (Time) panel

Use the Distributed Time Service (DTS) to synchronize the different time clocks of the systems within a cell. Select this option if you want to accept time from DCE DTS time servers. Click on the appropriate DTS button in the Configuration panel to configure your system as a **DTS client**, a **DTS local server**, or a **DTS global server**.

To enable cell time synchronization select **Automatically synchronize local time with cell time**. If you select the cell time option, you must also specify the host identification of the desired DTS server.

If your cell spans multiple local area networks (LANs), you must specify the name of your LAN in the **LAN profile name** field.

When you have completed your selections, click on **Next**. The **Full DCE Client Configuration** panel displays a summary of your configuration information.



Figure 7. The next Full DCE Client Configuration (Summary) panel

If the information is incorrect, click on **Back** to return to the appropriate panel to correct the information. If the information is correct, click on **Finish** to configure your system. The **DCE Configuration** panel is displayed with your configuration status information.

### Optional QSHHELL Command

You can also configure a full client by issuing the following command at the QSHHELL command line:

```
config.dce -config_type full [-cell_name cell_name] [-dce_hostname dce_hostname]
[-cell_admin cell_admin_id][-sec_master security_server] [-cds_server cds_server]
[-lan_profile profile] client_components
```

See the *IBM DCE Administration Commands Reference* for more information about this command.

## Configuring a Local Client

To configure your system as a local DCE client, from the **DCE Configuration** panel (see Figure 4 on page 14):

- Click on the **Configuration** menu.
- Click on **Create**.
- Click on **Local Client**.

The **Local DCE Client Configuration** panel is displayed.

The screenshot shows a configuration window titled "Local DCE Client Configuration - xyz.host". It features several input fields and checkboxes. The "Cell name" field is empty. The "Local host" section includes "DCE host name" (empty) and "Host identification" (filled with "rchas14d.rchland.ibm.com"). The "Servers" section includes "Security Master server host identification" (empty) and "CDS server host identification" (empty). The "Time" section includes an unchecked checkbox "Automatically synchronize local time with cell time" and "DTS server host identification" (empty). At the bottom, there is an unchecked checkbox "Start when TCP/IP is started" and four buttons: "< Back", "Next >", "Cancel", and "Help".

Figure 8. The Local DCE Client Configuration panel

Enter the cell name in the required **Cell name** field.

To enable cell time synchronization, select **Automatically synchronize local time with cell time**. If you select the cell time option, you must also specify the host identification of the desired DTS server.

To start DCE automatically, select **Start when TCP/IP is started**.

When you have completed your selections, click on **Next**. The **Local DCE Client Configuration** panel displays a summary of your configuration information.

If the information is incorrect, click on **Back** to return to the appropriate panel



Figure 9. The next Local DCE Client Configuration (Summary) panel

to correct the information. If the information is correct, click on **Finish** to configure your system. The **DCE Configuration** panel is displayed with your configuration status information.

### Optional QSHHELL Command

You can also configure a local client by issuing the following command at the QSHHELL command line:

```
config.dce -config_type local [-cell_name cell_name] [-dce_hostname dce_hostname]  
[-sec_master security_server] [-cds_server cds_server] client_components
```

See the *IBM DCE Administration Commands Reference* for more information about this command.



---

## Changing Your System Configuration

DCE for AS/400 provides a way to easily change whether to autostart DCE on your system without going through the entire configuration process again.

With the **Modify** option on the **Configuration** menu, you can change your system configuration by enabling or disabling the following option:

- Start when TCP/IP is started

To change the configuration of your system:

1. From the **DCE Configuration** panel, click on the **Configuration** menu and click on **Modify**. The Modify DCE Configuration panel is displayed.

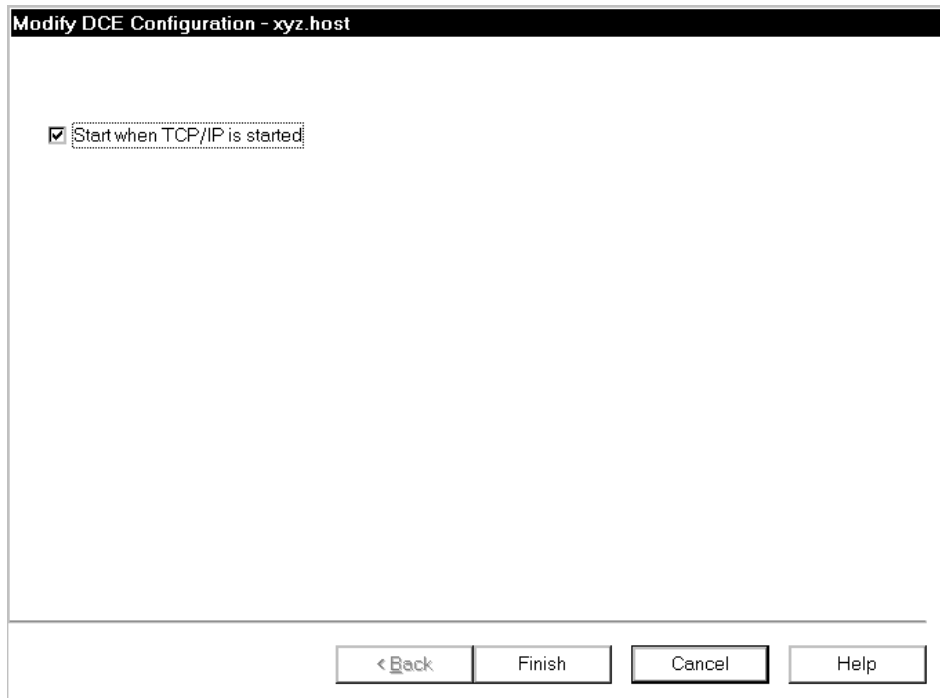


Figure 10. The Modify Configuration panel

2. Select **Start when TCP/IP is started**.
3. Click on **Finish** to change your configuration and return to the **DCE Configuration** panel.

**Note:** To make other configuration changes to your system, you must first unconfigure your system before reconfiguring it. See “Unconfiguring Your System” on page 22 for information about unconfiguring.

---

## Unconfiguring Your System

The DCE configuration component of Operations Navigator enables you to delete the DCE configuration from your AS/400 system. You can perform either a Full client unconfiguration or a Local client unconfiguration. For both unconfiguration types, use the **Delete** option to delete the configuration.

A Full client **Delete** removes all references to the client from the cell database. This includes objects in the CDS namespace and all security account information in the security registry. It also removes local configuration files. You need the cell administrator's principal name and password for this option.

A Local client **Delete** removes only local configuration files from the AS/400. In order to fully remove the host from the cell, perform an **admin** unconfiguration for the host from another machine in the cell. For more information about **admin** unconfiguration, see "Chapter 4. Setting Up DCE for AS/400" on page 9.

After unconfiguration, you must reconfigure the system before you can use the DCE services.

If you wish to selectively unconfigure DCE components, you must use the QSHELL command line. You cannot use the DCE configuration GUI. See the Optional QSHELL Command sections of "Full Client Unconfiguration" and "Local Client Unconfiguration" on page 24 for more information.

During unconfiguration, the actions that occur in order to delete the permanent configuration data are logged in the session log file.

---

## Full Client Unconfiguration

If you decide to discard your current configuration:

1. Click on the **Configuration** menu, and then click on **Delete**.
2. Click on **Full Client**. A dialog box box is displayed.

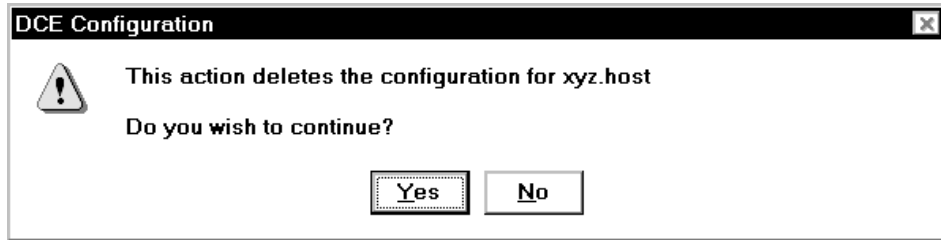


Figure 11. The DCE AS/400 Configuration dialog box

You are asked to confirm this action.

3. Click on **Yes**. The **Delete DCE Configuration** panel is displayed:



Figure 12. The Delete DCE Configuration panel

4. Type the correct information in the **Principal account name** and the **Principal password** fields.
  5. Click on **OK** to delete the configuration from both the host and the cell.
- The **DCE Configuration** panel is now displayed.

### Optional QSHHELL Command

You can also unconfigure a full client by issuing the following command at the QSHHELL command line:

```
unconfig.dce -config_type full [-cell_admin cell_admin id] [-dependents] [-force] components
```

See the *IBM DCE Administration Commands Reference* for more information about this command.

---

## Local Client Unconfiguration

To delete your configuration from the host using **Delete**:

1. Click on the **Configuration** menu, and then click on **Delete**.
2. Click on **Local Client**. A dialog box is displayed.  
You are asked to confirm this action.

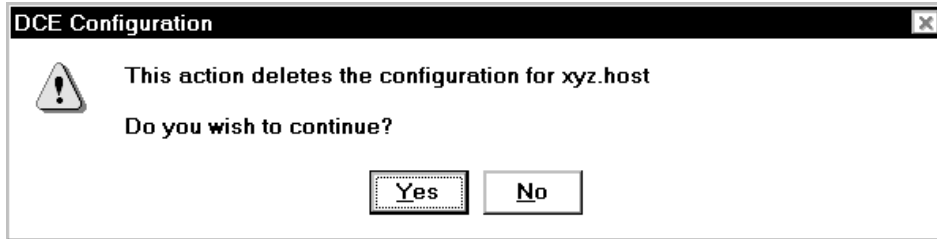


Figure 13. The DCE AS/400 Configuration dialog box

3. Click on **Yes** to delete the configuration from the host.  
The **DCE Configuration** panel is now displayed.

### Optional QSHHELL Command

You can also unconfigure a local client by issuing the following command at the QSHHELL command line:

```
unconfig.dce -config_type local [-dependents] [-force] components
```

See the *IBM DCE Administration Commands Reference* for more information about this command.

**Note:** To completely remove the host, you must perform an **admin** unconfiguration from another machine in the cell.

---

## Checking the Configuration Status

During the configuration process, the **DCE Configuration** panel enables you to view the status of the various DCE components. The **DCE Configuration** panel shows the various components, their current configuration status, and their running status. This panel is automatically updated after each configuration operation.

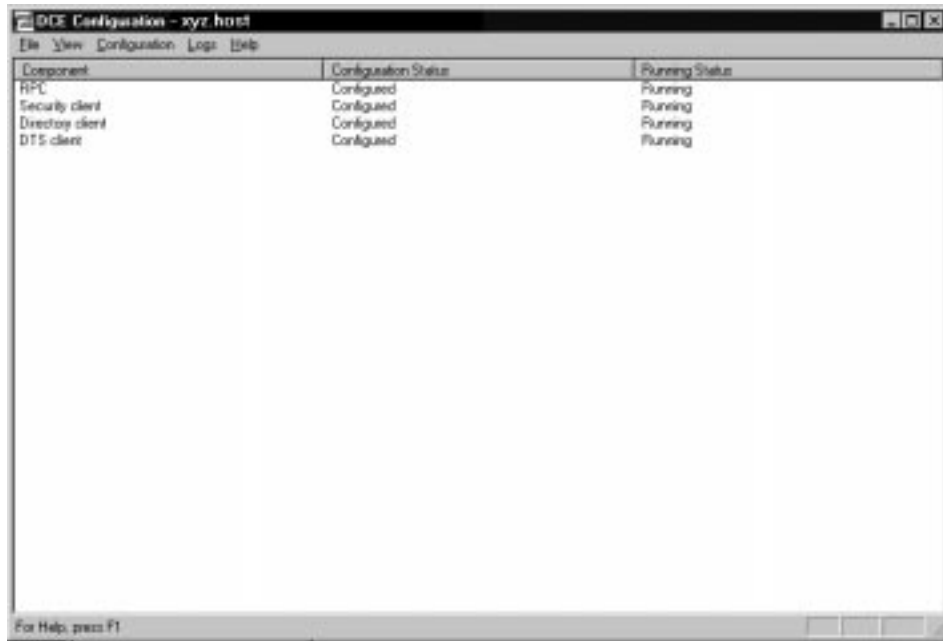


Figure 14. The DCE Configuration panel

You can also click on **View** and then click on **Refresh** to update the status.

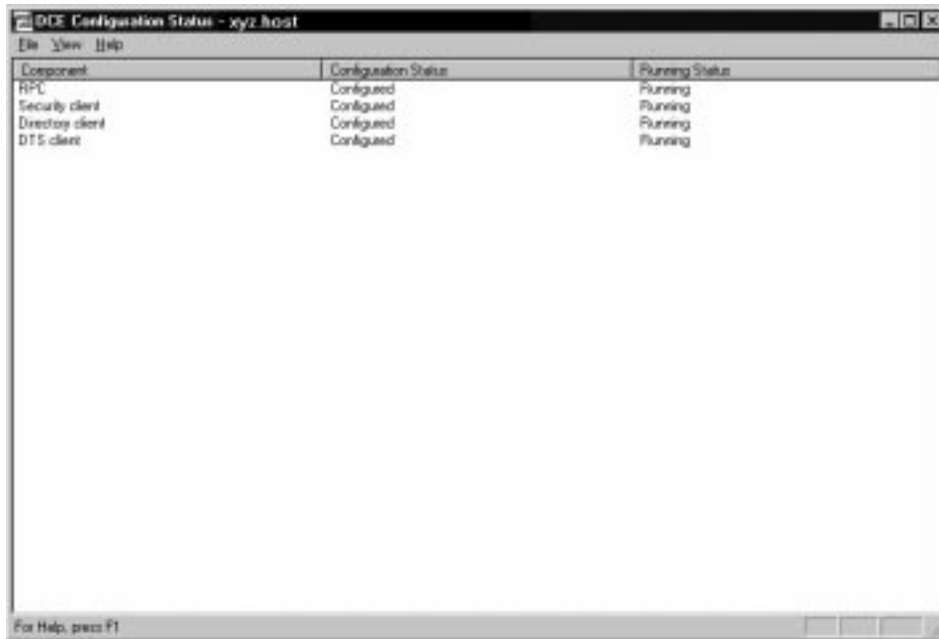


Figure 15. The DCE Configuration (Status) panel

**Note:** If you are not configuring DCE, you can check the status of the DCE components through the **Operations Navigator** folder. See “Running DCE Configuration” on page 12. After double-clicking on **TCP/IP**, right-click on **DCE** and select **Status** to display the **DCE Configuration Status** panel. This panel is not automatically updated. Click on **View** and then click on **Refresh** to update the status.

### Optional QSHHELL Command

You can also show a local client configuration by issuing the following command at the QSHHELL command line:

```
show.cfg [all] [dce] [usage] [-?] [help] [operations]
```

See the *IBM DCE Administration Commands Reference* for more information about this command.

---

### Viewing the Log File

Events that have occurred during a DCE configuration session are logged in a file. This file is created automatically by DCE configuration and is written to every time you run a DCE configuration.

The **Full DCE Configuration Log** (full log) contains up to the last 100,000 bytes that are logged by DCE configuration commands. The **Last DCE Configuration Log** (last log) contains information from the last DCE configuration command that was run on the AS/400 machine.

The file location for the full log file is:

**/opt/dcelocal/etc/cfgdce.log**

The file location for the last log file is:

**/opt/dcelocal/etc/cfgdce.run**

When the full log file reaches a size of 100,000 bytes, the file is renamed to **/opt/dcelocal/etc/cfgdce.bck** on the AS/400. A new DCE configuration full log is started.

**Note:** If you want to increase the size of the log file, use the environment variable **DCE\_CFG\_LOG\_MAX** on the AS/400 to change the byte limit of the log.

To view the last log file:

1. From the DCE Configuration panel, click on **Logs**.
2. Click on **View Last** to display the **Last DCE Configuration Log** panel.



Figure 16. The Last DCE Configuration Log panel

3. Click on **Refresh** to update the log file or **Cancel** to return to the **DCE Configuration** panel.

To view the Full log file:

1. From the DCE Configuration panel, click on **Logs**.
2. Click on **View Full** to display the **Full DCE Configuration Log** panel.



Figure 17. The Full DCE Configuration Log panel

3. Click on **Refresh** to update the log file or **Cancel** to return to the **DCE Configuration** panel.

## Viewing and Correcting Errors

If DCE configuration encounters errors, it writes error messages to the log files and tries to continue the configuration. If DCE configuration encounters an irrecoverable error, it stops the configuration. As part of the configuration procedure, DCE configuration tries to start the DCE services needed to support your configuration. If DCE configuration encounters an irrecoverable error while starting DCE services, the services do not start. The associated configuration files remain in the state that they were in immediately before the irrecoverable error occurred.

The following topics describe how to deal with errors that are encountered during configuration.

### Retry the Operation



Some errors occur when DCE configuration data is damaged because of a transitory condition in the environment. The error might not recur if you retry the operation. If DCE configuration encountered errors during configuration, you must first unconfigure your system. For the unconfiguration operation, see “Unconfiguring Your System” on page 22. After your system has been unconfigured, click on the **Configuration** menu, choose the same configuration option as before, and try the configuration again.

### Checking the Log Files

The log files provide a valuable record of the information that you specified during configuration. Depending on your configuration, you might want to check one or more of the following:

- Cell name
- Host identification
- Security server identification
- CDS server identification

See “Viewing the Log File” on page 26 for more information.

**Note:** Ensure that you have used a login principal name and password with the necessary DCE privileges to perform the configuration.

### Make Sure that the DCE Servers Are Running

To make sure that the DCE servers are running, you must have access to the system on which a particular server should be running. Contact the DCE cell administrator for information on server status. If you are a cell administrator, refer to the server documentation for information on how to show server status.

---

## Starting, Stopping, and Cleaning Up DCE

You can start, stop, or clean up DCE by using the Operations Navigator or the QSHELL command line. You can also start or stop DCE from the CL command line.

### Operations Navigator

To use the Operations Navigator while connected using the QDCE user profile, perform the following:

1. Double-click on the desired machine.
2. Double-click on **Network**.

3. Double-click on **Servers**.
4. Double-click on **TCP/IP**.
5. Right-click on **DCE**.

The pop-up menu is displayed.

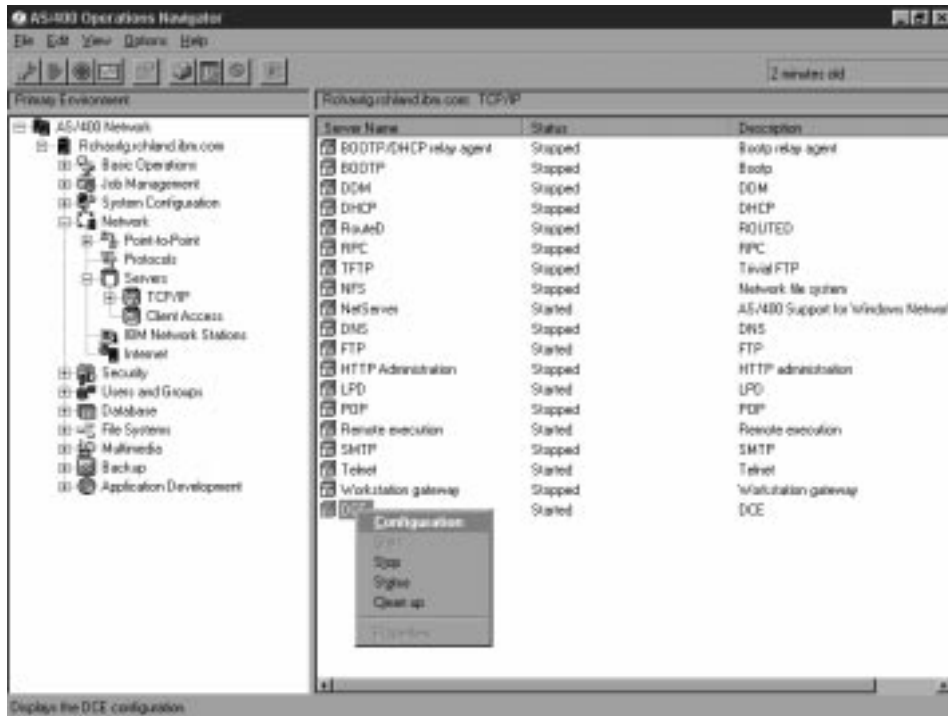


Figure 18. The Operations Navigator panel

**Note:** You need to use the QDCE profile to perform these operations. If you are not signed on as QDCE, you will only have the **Status** selection displayed on the context menu.

Select **Start**, **Stop**, or **Clean up**. The **Start** and **Stop** selections will be enabled or disabled depending on the state of DCE on your system. In the previous figure DCE is already running, so the **Start** selection is disabled.

**Note:** When you select **Start** or **Stop**, the processing returns to the Operations Navigator before the completion of the request. Click on **View** and then click on **Refresh** to update the status to determine when your request has completed.

If you encounter problems when starting DCE, select **Clean up**. This function deletes re-creatable files that are used internally by DCE daemons during

runtime for saving data, credentials, and cache information. These files will be automatically re-created when you start DCE again.

### **QSHLL Command Line**

You can optionally issue the following commands at the QSHLL command line:

**start.dce** — starts DCE on your system.

**stop.dce** — stops DCE on your system.

**clean\_up.dce** — cleans the DCE databases, sockets, and cache files, creates backup log files, and removes DCE-generated core files.

### **CL Command Line**

You can optionally issue the following commands at the CL command line:

**STRTCPSVR SERVER(\*DCE)** — starts DCE on your system.

**ENDTCPSVR SERVER(\*DCE)** — stops DCE on your system.



---

## Chapter 6. DCE Features and Functions

This section describes the functions that are included in this release of DCE Base Services for AS/400. It also provides lists of supported DCE commands and supported **dcecp** commands.

---

### Supported Functions in this Release

DCE Base Services for AS/400 provide support for remote procedure calls, the client functionality for cell directory service and security, time, message handling, and serviceability. This release of DCE for AS/400 supports the following base components and functions:

- The **Remote Procedure Call (RPC)** function enables you to create and run client and server applications. The RPC runtime service puts into effect the network protocols by which the client and server sides of an application communicate.
- **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 Universal Time Coordinated (UTC), an international time standard.
- **DCE Security Client** supports RPC runtime libraries, and some security operations within those libraries.
- **DCE Cell Directory Client** is a distributed, replicated database service that is used to store names and attributes of resources that are located in a DCE cell.
- **Single Threaded RPC Client** is a single sequential flow of control within a program. It is the active running of a designated routine, including any nested routine calls. Within a single thread, there is a single point of operation. Most traditional programs consist of a single thread. Threads are lightweight processes that share a single address space. Each thread shares all the resources of the originating process, including signal handlers and descriptors. Each thread has its own thread identifier, scheduling policy and priority, errno value, thread-specific data bindings, and the required system resources to support a flow of control.
- **Multithreaded Programming Environment** is a division of a program into multiple threads (parts) that run concurrently.
- **GSSAPI Extensions** — GSSAPI extensions are a set of application program interfaces (APIs) that provide non-RPC applications the ability to use the DCE security authentication protocol. Use the GSSAPI to establish credentials or to extract Extended Privilege Attribute Certificates (EPAC) for a non-RPC application.

- **CDMF** (Common Data Masking Facility) provides support for primarily non-North American customers to encrypt application data through the RPC programming interface.
- **An Application Developer's Tool Kit** allows client and server programs that are written in C to use DCE RPC in a highly transparent manner. See "Appendix C. DCE Base Services for AS/400 Application Development Kit" on page 59 for more information.

**Note:**

The DCE 2.2 for AIX documents that are provided with this product discusses these functions in great detail . See "Appendix B. Documentation Provided with this Release" on page 55 for more information.

## Supported DCE and DCECP Commands

The following DCE and dcecp commands are available through the QSHELL interface. To start an interactive QSHELL session, enter **QSH** (or **STRQSH**) CL command.

### Supported DCE Commands

The following DCE commands are available through the QSHELL interface.

<b>dcecp</b>	<b>kinit</b>	<b>rmxcred</b>
<b>dce_login</b>	<b>klist</b>	<b>uuidgen</b>
<b>kdestroy</b>		

**Note:** The following commands, under the QSHELL have the following known limitations:

**dcecp** - The command prompt is displayed by using the **-tty** option.

**dce\_login** - The password is displayed on the screen.

**Attention:** Because the password is displayed on the screen, use extreme care when issuing the **dce\_login** command to avoid compromising system security.

For more information about the individual commands, see the *IBM DCE Administration Commands Reference*.

### Supported DCE Configuration Commands

The QSHELL command line supports the following DCE configuration commands:

**clean\_up.dce**  
**config.dce**  
**start.dce**

**stop.dce**  
**unconfig.dce**

**Attention:** Because the password is displayed on the screen, use extreme care when issuing the **config.dce** and **unconfig.dce** commands to avoid compromising system security.

For more information about the individual commands, see the *IBM DCE Administration Commands Reference*.

### Supported DCECP Commands

The QSHELL supports a minimal set of **dcecp** commands for this release. Commands that are used to administer the cell will be issued from other clients in the cell. Other DCE commands provide the ability to log in, to logout, or to create uuids. See “Supported DCE Commands” on page 34 for a listing of these commands.

The following are the only supported **dcecp** commands:

Table 1. Supported **dcecp** commands

<b>account</b>	<b>endpoint</b>	<b>object</b>	<b>rpcentry</b>
create	show	create	create
delete		delete	export
modify		show	show
show			
<b>acl</b>	<b>group</b>	<b>principal</b>	<b>secval</b>
modify	add	catalog	activate
show	create	create	deactivate
	delete	delete	status
	list	show	update
	show		

Table 1. Supported **dcecp** commands (continued)

<b>cdscache</b>	<b>keytab</b>	<b>registry</b>	<b>server</b>
create	add	delete	create
	create	dump	delete
	delete	show	show
	list		start
	show		stop
<b>directory</b>	<b>organization</b>	<b>rpcprofile</b>	<b>user</b>
create	add	add	create
delete	list	create	delete
list		show	show
show			
synchronize			

For more information about the individual commands, see the *IBM DCE Administration Commands Reference*.



---

## Chapter 7. Serviceability and Internationalization

The following section discusses X/Open Portability Guidelines, Issue 4 (XPG4) and coded character set identifier (CCSID) considerations. It provides a conversion chart for CCSIDs. It also provides a sample message handling program for setting up the XPG4 or Portable Operating System Interface for Computer Environments (POSIX) locale.

---

### XPG4 Internationalization Considerations

#### XPG4 Versus AS/400-Proprietary Internationalization

DCE uses the XPG4 locale model for internationalization. This programming model works correctly only when all code that is running in the same process is using the XPG4 model. Ideally, for maximum consistency of character data handling, all software that interacts should be using the XPG4 model.

AS/400 supports XPG4 programming that use the POSIX locales (locale objects of type \*LOCALE). However, like many other operating systems, AS/400 also supports proprietary methods for internationalization. For the most part, these proprietary methods and XPG4 are unaware of each other. A program that mixes the two models may act in an inconsistent fashion because it is using multiple values for the same piece of state information, such as CCSID or code page.

AS/400 provides the capability to partially synchronize the AS/400-proprietary national language support (NLS) state information with the XPG4 state information. For this release, DCE requires the use of these techniques instead of the standard XPG4 techniques that are documented in the AS/400 compiler books and in the XPG4 standard. See “Summary of NLS Setup Considerations and Programming Considerations” on page 42 for an example.

#### XPG4 Locale and Message Catalogs: NLSPATH Environment Variable

DCE uses XPG4 message catalogs for all DCE error and informational messages. In addition, the DCE Messaging and Serviceability APIs provide a convenient abstraction of the XPG4 message handling functions for use by DCE applications that support multiple language translations. See the chapters on Messaging and Serviceability in *IBM DCE for AIX, Version 2.2: Application Development Guide—Core Components* and the **sams** command in *IBM DCE for AIX, Version 2.2: Administration Commands Reference*.

Locale names are used to locate the message catalogs. DCE message catalogs are located in directories of the form `/opt/dcelocal/nls/msg/xx/yy/` where `xx` is the language portion of the locale name and `yy` is the territory (or territory and CCSID). For example, the directory `/opt/dcelocal/nls/msg/EN/US/` contains the message catalogs for English. If your application was translated into French using CCSID 297, you could install your message catalogs in `/opt/dcelocal/nls/msg/FR/FR`. If you additionally required French using CCSID 500, you could create a locale called `FR_FR500`. See *AS/400 International Application Development*, and install the catalogs in `/opt/dcelocal/nls/msg/FR/FR500` for more information.

The environment variable `NLSPATH` specifies the path that is searched when the XPG4 runtime code looks for message catalogs. The `NLSPATH` variable can also specify a series of paths that are separated by colons (:). For correct functioning of DCE messages, you must include this clause in the `NLSPATH`:  
`/opt/dcelocal/nls/msg/%l/%t/%N`

**Note:** PTF SF49359 is necessary to create the directory structure for the DCE message catalogs to support this `NLSPATH` clause.

This path uses three standard XPG4 substitution variables, `%l`, `%t`, and `%N`. Case is significant. The language portion of the locale name is substituted for `%l`. The territory/CCSID portion is substituted for `%t`, and a message catalog name is substituted for `%N`. As with the program path, if multiple XPG4 applications are being used in the same environment and their message catalogs are installed in different locations, the `NLSPATH` may consist of multiple clauses, separated by a colon (:).

Returning to the French example, if a user of your application sets the locale to `FR_FR.LOCALE` (see “Summary of NLS Setup Considerations and Programming Considerations” on page 42), the DCE message handling and serviceability APIs will use the `NLSPATH` to find your French message catalogs, which are installed in `/opt/dcelocal/nls/msg/FR/FR`. Since DCE itself has not been translated into French in this release, DCE will not find its own catalogs in this directory. It will use default English messages instead. DCE has been translated into Italian. If a user sets locale to Italian, and if your application has installed Italian message catalogs in `/opt/dcelocal/nls/msg/IT/IT`, that user will see both your messages and DCE messages in Italian. You never need to change the DCE clause of the `NLSPATH`, because it uses the substitution variables to find the correct set of catalogs for the current locale.

---

## Network CCSID for DCE Communication

This release of DCE includes a Remote Procedure Call (RPC) feature which allows DCE applications to specify automatic code set conversion for certain types of RPC data. See "Writing Internationalized RPC Applications" in *IBM DCE for AIX, Version 2.2: Application Development Guide—Core Components* for a detailed description.

In addition, DCE internal data that flows between various DCE clients and servers must be converted between extended binary-coded decimal interchange code (EBCDIC) and American National Standard Code for Information Interchange (ASCII). Because the AS/400 DCE EBCDIC client does not know the code set of the server, it is necessary to "guess" the correct encoding for data flowing to and from the server. The AS/400 DCE client code selects the default network ASCII CCSID that is based on a best fit with the current XPG4 EBCDIC CCSID of the client process. The tables that follow list these default values.

These are not the only valid combinations. DCE allows you to override the default network CCSID by setting the environment variable **DCE\_NETWORK\_CCSID** to the new CCSID, using the format that is used for AS/400 **iconv()** parameters. If, for example, your AS/400 client is running in CCSID 037 and you want to run your AIX or NT server in code page 850, you can set **DCE\_NETWORK\_CCSID** to **IBMCCSID00850** before starting DCE. If you override the DCE default, make sure to check that AS/400 has a conversion table to support the new combination. The *AS/400 International Application Development* publication documents the AS/400 CCSID conversion tables.

ASCII DCE nodes do not do CCSID conversion. If you change the network CCSID, make sure that it is usable, as is, by the ASCII DCE installation.

Table 2. CCSID Conversion Tables

EBCDIC CCSID	DCE Network Default CCSID	Description of Network CCSID
037	819	ISO 8859-1; Latin-1
256	819	ISO 8859-1; Latin-1
273	819	ISO 8859-1; Latin-1
277	819	ISO 8859-1; Latin-1
278	819	ISO 8859-1; Latin-1
280	819	ISO 8859-1; Latin-1
284	819	ISO 8859-1; Latin-1
285	819	ISO 8859-1; Latin-1
297	819	ISO 8859-1; Latin-1
500	819	ISO 8859-1; Latin-1

Table 2. CCSID Conversion Tables (continued)

<b>EBCDIC CCSID</b>	<b>DCE Network Default CCSID</b>	<b>Description of Network CCSID</b>
871	819	ISO 8859-1; Latin-1

Table 3. CCSID Conversion Tables

<b>EBCDIC CCSID</b>	<b>DCE Network Default CCSID</b>	<b>Description of Network CCSID</b>
420	1089	ISO 8859-6; Arabic
8612	864	PC-Data; Arabic
424	916	ISO 8859-8; Hebrew
870	912	ISO 8859-2; Latin-2
423	813	ISO 8859-7; Greek/Latin
875	813	ISO 8859-7; Greek/Latin
918	868	PC-Data; Urdu

Table 4. CCSID Conversion Tables

<b>EBCDIC CCSID</b>	<b>DCE Network Default CCSID</b>	<b>Description of Network CCSID</b>
880	915	ISO 8859-5; Cyrillic, 8-bit
1025	915	ISO 8859-5; Cyrillic, 8-bit
905	920	ISO 8859-9; Latin 5 (ECMA-128, Turkey TS-5881)
1026	920	ISO 8859-9; Latin 5 (ECMA-128, Turkey TS-5881)
1097	1098	Farsi - PC
1112	921	Baltic, 8-bit
1122	922	Estonia, 8-bit
1123	1125	Cyrillic Ukraine PC-Data
1130	1258	MS Windows, Vietnamese
1132	1133	ISO-8, Lao

Table 5. CCSID Conversion Tables

<b>EBCDIC CCSID</b>	<b>DCE Network Default CCSID</b>	<b>Description of Network CCSID</b>
836	903	Simplified Chinese, PC-Data, SBCS
13124	903	Simplified Chinese, PC-Data, SBCS
837	928	Simplified Chinese, PC-Data, DBCS 1880 UDC
4933	1380	Simplified Chinese, PC-Data, DBCS 1880 UDC, IBM GB
935	1381	Simplified Chinese, PC-Data, DBCS 1880 UDC, IBM GB, SAA
1388	1381	Simplified Chinese, PC-Data, DBCS 1880 UDC, IBM GB, SAA

Table 6. CCSID Conversion Tables

EBCDIC CCSID	DCE Network Default CCSID	Description of Network CCSID
836	903	Simplified Chinese, PC-Data, SBCS
13124	903	Simplified Chinese, PC-Data, SBCS
837	928	Simplified Chinese, PC-Data, DBCS 1880 UDC
4933	1380	Simplified Chinese, PC-Data, DBCS 1880 UDC, IBM GB
935	1381	Simplified Chinese, PC-Data, DBCS 1880 UDC, IBM GB, SAA
1388	1381	Simplified Chinese, PC-Data, DBCS 1880 UDC, IBM GB, SAA

Table 7. CCSID Conversion Tables

EBCDIC CCSID	DCE Network Default CCSID	Description of Network CCSID
835	927	Traditional Chinese, PC-Data, DCBS, 6204 UDC
937	950	Traditional Chinese, PC-Data, mixed, IBM BIG-5
28709	950	Traditional Chinese, PC-Data, mixed, IBM BIG-5

Table 8. CCSID Conversion Tables

EBCDIC CCSID	DCE Network Default CCSID	Description of Network CCSID
290	897	Japanese, PC-Data, SBCS
300	301	Japanese, PC-Data, DBCS, 1880 UDC
4396	301	Japanese, PC-Data, DBCS, 1880 UDC
930	932	Japanese, PC-Data, mixed, 1880 UDC
939	932	Japanese, PC-Data, mixed, 1880 UDC
5026	932	Japanese, PC-Data, mixed, 1880 UDC
5035	932	Japanese, PC-Data, mixed, 1880 UDC
1027	942	Japanese, PC-Data, mixed, 1880 UDC, SAA

Table 9. CCSID Conversion Tables

EBCDIC CCSID	DCE Network Default CCSID	Description of Network CCSID
833	949	IBM KS Code, PC-Data, mixed, 1880 UDC
933	949	IBM KS Code, PC-Data, mixed, 1880 UDC
834	926	Korean, PC-Data, DBCS, 1880 UDC

Table 10. CCSID Conversion Tables

EBCDIC CCSID	DCE Network Default CCSID	Description of Network CCSID
838	874	Thai, PC-Data, SBCS, extended
9030	9066	Thai, PC-Data, SBCS, extended, 10 added

---

## Summary of NLS Setup Considerations and Programming Considerations

### Setting the XPG4/POSIX Locale

In this release, the XPG4 process CCSID must match the AS/400-proprietary process CCSID. To ensure this, you must set the locale in the system or user profile. You must set the \*ccsid job attribute, and you must not use the documented XPG4 techniques for setting or changing the locale.

The following example shows an AS/400 command that will set the locale to Italian:

```
CHGUSRPRF USRPRF(user_name) SETJOBATR(*ccsid) LOCALE('/QSYS.LIB/IT_IT.LOCALE')
```

### Sample Messaging Program

The **hello\_svc** example program in the DCE Application Toolkit shows how to build an application that uses XPG4 message catalogs by using the IBM Visual Age Compiler.

In addition, your XPG4 programs should include the following call to **setlocale()** at the beginning of the executable code:

```
-----code sample-----
#include <locale.h>
int
main()
{
    /* declarations */
    .
    .

    /* start of executable code */
    setlocale(LC_ALL, "");
    .
    .
}
-----
```

### Network CCSID

If the DCE default network CCSID is not correct for your environment, you can override it by setting this environment variable:

```
DCE_NETWORK_CCSID = "IBMCCSIDnnnnn"
```

where nnnnn is the right-justified, 0-filled CCSID number.

## NLSPATH

Make sure the NLSPATH is set to, or contains the clause:

```
/opt/dce1ocal/n1s/msg/%1/%t/%N
```





---

## Appendix A. Helpful Tips

This section provides information that will help you with some of the more common problems that you might encounter. See the **Problem Prevention**, **Problem Determination**, **Problem Resolution**, and **Serious Problems** sections of the *IBM DCE for AIX, Version 2.2: Problem Determination Guide* for additional helpful information.

---

### Threads

DCE threads are based on the Portable Operating System Interface for Computer Environments (POSIX) 1003.1c, draft 4 standards. The AS/400 kernel threads provide pthread APIs from the POSIX 1003.1c, approved standard (draft 10). The differences in thread draft levels have been resolved so that calls are mapped correctly. There are two conditions that you need to be aware of, however. The `pthread_getprio()` and the `pthread_attr_getprio()` application program interfaces (APIs) could return a `-1` respectively as the current priority and the defined priority of a thread. If a `-1` is returned, check the error number (`errno`) for the corresponding error value. See <http://www.softmall.ibm.com/as400/threads> for more information about `errno`s. If the `errno` is zero (`0`), then the `-1` is a valid AS/400 thread priority value, that is in the range `0` to `-99`.

**Note:** The `dce\pthread.h` header file, that provides the mapping for the calls, is located in the `dce` subdirectory. In order to get the DCE threads functionality, all DCE application program files need to include the header file `# include <dce\pthread.h>` instead of the `# include <pthread.h>` header file.

---

### Running Multithreaded AS/400 DCE Applications

Multithreaded AS/400 DCE applications are compiled using `_MULTI_THREADED` or by calling `pthread_create`. If the program is run from the AS/400 command line, the syntax is `spawn pgm(libname/pgmname) parm("parm1" "parm2" "parm3" ...)`. For example:

```
spawn pgm(perf/server) parm("parm1" "parm2" ...)
```

**Note:** The syntax call `pgm(libname/prmname) parm(...)` is for old single-threaded AS/400 DCE applications.

If the program is run from QSH mode, type `fullpathname/pgmname....` For example:

```
/QSYS.LIB/PERF.LIB/server.pgm parm1 parm2 ...
```

Debug multithreaded AS/400 DCE applications are compiled with **DEBUG** and **\_MULTI\_THREADED** from the AS/400 command line. If you want to enable debug mode, you must add **DEBUG(\*YES)** after **parm** in the multithreaded syntax. For example:

```
spawn pgr(perf/server) parm("parm1""parm2") DEBUG(*YES)
```

---

## The DCE `dce_free` Routine

Use the `dce_free` routine to free allocated memory that is returned by some of the DCE service functions or is allocated by the application code that includes `dce\pthread.h`. All DCE routines allocate memory by using a DCE version of `malloc`, `dce_malloc`, instead of the C runtime routine `malloc`. When the memory is no longer needed, you need to free it. Typically, calling the C free routine does this. However, on AS/400 platforms, unpredictable results can occur using `free`, if the application code does not include `dce\pthread.h`. You must use the `dce_free` routine to replace the standard C free routine. The `dce_free` routine is an entry point within the DCE service programs that call the C free routine.

### `dce_free`

#### Purpose

Frees memory that is allocated by `dce_malloc`, if the application code does not include `dce\pthread.h`.

#### Format

```
#include <dce\pthread.h>
void dce_free(
    void *ptr);
```

#### Parameters

##### Input

**ptr** The block of storage to be freed. This block of storage must have been allocated by a call to one of the DCE routines that are listed in **Usage**.

**Usage** The `dce_free` routine is used on the AS/400 platforms to free memory that is allocated by the following DCE routines:

<code>dce_msg_get</code>	<code>dce_cf_get_cell_name</code>
<code>dce_msg_get_msg</code>	<code>dce_cf_get_host_name</code>
<code>dce_sprintf</code>	<code>dce_cf_dced_entry_from_host</code>
<code>dce_pgm_sprintf</code>	<code>dce_cf_get_csrgy_filename</code>
<code>dce_aud_print</code>	<code>dce_db_header_fetch</code>
<code>dce_cf_find_name_by_key</code>	

**Notes** This routine is not a general routine for freeing memory, nor is it a replacement for other DCE routines that free memory (for example, **rpc\_string\_free**). Ensure that any calls to **dce\_free** are within the bounds of an **#ifdef** for portability.

### Examples

This example shows the use of **dce\_free** within an **#ifdef**.

```
#ifdef IBMAS4NT
# include <dce\pthread.h>
#else
# include <pthread.h>
#endif
dce_free (ptr);
```

### Related Information

<b>dce_msg_get</b>	<b>dce_cf_get_cell_name</b>
<b>dce_msg_get_msg</b>	<b>dce_cf_get_host_name</b>
<b>dce_sprintf</b>	<b>dce_cf_dced_entry_from_host</b>
<b>dce_pgm_sprintf</b>	<b>dce_cf_get_csrgy_filename</b>
<b>dce_aud_print</b>	<b>dce_db_header_fetch</b>
<b>dce_cf_find_name_by_key</b>	

---

## Remote Procedure Call (RPC) Memory Management

When called to handle a remote operation, RPC client stubs allocate and free memory by using whatever memory management scheme is currently in effect. Client code is generic code that can be called from either RPC clients or RPC servers. It can use DCE RPC stub support routines to control which memory management scheme the stubs will use.

If client code has not explicitly set the memory management routines, the RPC client stubs use the following defaults:

- When called from manager code, and the operation contains one or more parameters that are full or unique pointers, the client stubs use the **rpc\_ss\_allocate** and **rpc\_ss\_free** routines.
- When called from manager code, and the ACF attribute **enable\_allocate** has been applied, the client stubs use the **rpc\_ss\_allocate** and **rpc\_ss\_free** routines.
- When called from any other context, the RPC client stubs use the operating system allocation and free routines, for example **malloc** and **free** on POSIX platforms.

Note that the memory management scheme established, whether explicitly or by default, is on a per-thread basis.

RPC server stubs do not allocate memory. Instead, they rely on the manager code, that is, the code that the server stubs call, to allocate it for them. The following give guidelines for how client code and manager code should use the various allocation and free routines that are provided with DCE.

**Note:** DCE provides two versions of DCE RPC stub support routines. The **rpc\_ss** routines raise an exception, while the **rpc\_sm** routines return an error status value. In all other ways, the routines are identical. Use the **rpc\_sm** routines instead of the **rpc\_ss** routines for compliance with Application Environment Specification for DCE RPC.

### Using **rpc\_ss\_allocate** and **rpc\_ss\_free** in Manager Code

Manager code uses either the **rpc\_ss\_allocate** and **rpc\_ss\_free** routines or the operating system allocation and free routines to allocate and free memory. Manager code uses **rpc\_ss\_allocate** to allocate storage for data that the server stub is to send back to the client. Manager code can either use **rpc\_ss\_free** to free the storage explicitly, or it can rely on the server stub to free it. After the server stub marshals the output parameters, it releases any storage that the manager code has allocated with **rpc\_ss\_allocate**.

Manager code can also use the **rpc\_ss\_free** routine to release storage pointed to by a full pointer in an input parameter. The freeing of the memory can be reflected on return to the calling application if the **reflect\_deletions** attribute has been specified as an operation attribute in the interface definition. See Interface Definition Language for instructions on how to declare the **reflect\_deletions** operation attribute.

Manager code uses the operating system's allocation routine to create storage for its internal data. The server stub does not automatically free memory that operating system's allocation routines have allocated. Instead, manager code must use the operating system's free routine to deallocate the memory explicitly before it exits.

When manager code makes a remote call, the default memory management routines are **rpc\_ss\_allocate** and **rpc\_ss\_free**.

### Using **rpc\_ss\_allocate** and **rpc\_ss\_free** in Client Code

Client code may also want to use the **rpc\_ss\_allocate** and **rpc\_ss\_free** routines as the stub memory management scheme. However, before client code can use **rpc\_ss\_allocate** and **rpc\_ss\_free**, it must first call the **rpc\_ss\_enable\_allocate** routine, which enables the use of **rpc\_ss\_allocate**. If client code calls **rpc\_ss\_enable\_allocate**, it must also call the **rpc\_ss\_disable\_allocate** routine before it exits its thread to disable use of **rpc\_ss\_allocate**. This routine releases all of the memory that is allocated by calls to **rpc\_ss\_allocate** in that thread since the call to **rpc\_ss\_enable\_allocate** was made. As a result, client code can

free each piece of allocated storage with **rpc\_ss\_free**. It can also have **rpc\_ss\_disable\_allocate** free the allocated storage all at once when it disables the **rpc\_ss\_allocate/free** memory management scheme.

Before calling **rpc\_ss\_enable\_allocate**, client code must ensure that it has not been called by code that has already set up the **rpc\_ss\_allocate/free** memory management scheme. As a result, if the client code can ensure that it has not been called from a manager routine, and that any previous calls to **rpc\_ss\_enable\_allocate** have been paired with calls to **rpc\_ss\_disable\_allocate**, it can safely call **rpc\_ss\_enable\_allocate**.

If client code cannot ensure that these conditions are true, it should check to make sure the **rpc\_ss\_allocate/free** scheme has not already been set up. For example:

```
/* Get RPC memory allocation thread handle */
rpc_ss_thread_handle_t thread_handle;
idl_void_p_t (*p_saved_alloc)(unsigned long);
void (*p_saved_free)(idl_void_p_t);
TRY
    thread_handle = rpc_ss_get_thread_handle();
CATCH(pthread_badparam_e)
    thread_handle = NULL;
ENDTRY
if (thread_handle == NULL) {
    /* Set up rpc_ss_allocate environment */
    rpc_ss_enable_allocate();
}
rpc_ss_swap_client_alloc_free(
    appl_client_alloc, appl_client_free,
    &p_saved_alloc, &p_saved_free);
```

After control returns from the client stub, the client code should again check to see whether **rpc\_ss\_allocate/free** has already been enabled before it calls **rpc\_ss\_disable\_allocate**:

```
rpc_ss_set_client_alloc_free(p_saved_alloc, p_saved_free);
/* If we set up rpc_ss_allocate environment, disable it now */
if (thread_handle == NULL)
    rpc_ss_disable_allocate();
```

## Using Your Own Allocation and Free Routines

At times it might be necessary for client code to change the routines that the client stubs use to allocate and free memory. For example, client code that is making an RPC call might want to direct the RPC client stubs to use special debug versions of malloc and free that check for memory leaks. Another example is an application that uses DCE RPC but needs to preserve its user's ability to free memory returned from the application using the platform's memory management scheme (rather than exposing the user to DCE). The **dce\_free** routine frees allocated memory. Some DCE routines use the C

runtime routine `malloc` to allocate memory. When these routines are finished with the allocated memory, you need to issue a routine to free the memory. It is important on AS/400 platforms, to use the `dce_free` routine. See “The DCE `dce_free` Routine” on page 46 for more information.

Client code that wants to use its own memory allocation and free routines can use the `rpc_ss_swap_client_alloc_free` routine to exchange the current client allocation and freeing mechanism for one supplied in the call. The routine returns pointers to the memory allocation and free routines formerly in use. Before calling `rpc_ss_swap_client_alloc_free`, client code must ensure that it has not been called from a manager routine.

Deallocation of allocated storage that is returned from the client stubs is not automatic. Therefore, client code must ensure that it uses the free routine that it specified in the call to `rpc_ss_swap_client_alloc_free` to deallocate each piece of allocated storage. Client code that swaps in memory management routines with `rpc_ss_swap_client_alloc_free` should use the `rpc_ss_set_client_alloc_free` routine before it exits to restore the old allocation and free routines.

## IDL Encoding Services Memory Management

Interface Definition Language (IDL) encoding-services stubs handle memory management in the same way as RPC client stubs. When you call an operation to which the `encode` or `decode` attribute, or both attributes have been applied, the encoding-services stub uses whatever client stub memory management scheme is currently in effect. See “Remote Procedure Call (RPC) Memory Management” on page 47 for further details on client stub memory management defaults and setting up memory management schemes.

You can control which memory management scheme the stubs will use by calling the `rpc_ss_swap_client_alloc_free` and `rpc_ss_set_client_alloc_free` routines. The first routine sets the memory management routines that are used by both the encoding and decoding stubs. The second routine restores the previous memory management scheme after encoding and decoding are complete.

**Note:** The memory management scheme established, whether explicitly or by default, is on a per-thread basis.

## Advanced Memory Management Support

Memory management can also involve setting and swapping the mechanisms that are used for allocating and freeing memory. The default memory

management routines are `malloc` and `free`. When the remote call occurs within manager code, the default memory management routines are `rpc_ss_allocate` and `rpc_ss_free`.

### Setting the Client Memory Mechanism

Use the `rpc_ss_set_client_alloc_free` routine to establish the routines that are used in allocating and freeing memory.

The syntax of the routine is:

```
void rpc_ss_set_client_alloc_free (
    idl_void_p_t (*p_allocate) (
        idl_size_t size),
    void (*p_free) (
        idl_void_p_t ptr)
);
```

The `p_allocate` parameter points to a routine that has the same procedure declaration as the `malloc` routine, and that was used by the client stub when performing memory allocation. The `p_free` parameter points to a routine that has the same procedure declaration as the `free` routine, and that the client stub used to free memory.

### Swapping Client Memory Mechanisms

This routine exchanges the current client allocation and freeing mechanism for one supplied in the call. The primary purpose of this routine is to simplify the writing of modular routine libraries in which RPC calls are made. To preserve modularity, any dynamically allocated memory returned by a modular routine library must be allocated with a specific memory allocator. When dynamically allocated memory is returned by an RPC call that is then returned to the user of the routine library, use `rpc_ss_swap_client_alloc_free` to make sure the desired memory allocator is used. Prior to returning, the modular routine library calls `rpc_ss_set_client_alloc_free` to restore the previous memory management mechanism.

The syntax of the routine is:

```
void rpc_ss_swap_client_alloc_free (
    idl_void_p_t (*p_allocate) (
        idl_size_t size),
    void (*p_free) (
        idl_void_p_t ptr),
    idl_void_p_t (**p_p_old_allocate) (
        idl_size_t size),
    void (**p_p_old_free) (
        idl_void_p_t ptr)
);
```

The *p\_allocate* parameter points to a routine that has the same procedure declaration as the malloc routine, and that was used by the client stub when performing memory allocation. The *p\_free* parameter points to a routine that has the same procedure declaration as the free routine and that was used by the client stub to free memory. The *p\_p\_old\_allocate* parameter points to a pointer to a routine that has the same procedure declaration as the malloc routine and that was used for memory allocation in the client stub. The *p\_p\_old\_free* parameter points to a pointer to a routine that has the same procedure declaration as the free routine and that was used for memory release in the client.

## Memory Management for Pointed-to Nodes

A full pointer can change its value across a call. Therefore, stubs must be able to manage memory for the pointed-to nodes. Managing memory involves allocating and freeing memory for user data structures.

### Allocating and Freeing Memory

Manager code within RPC servers usually uses the **rpc\_ss\_allocate** routine to allocate storage. Storage that is allocated by **rpc\_ss\_allocate** is released by the server stub after any output parameters have been marshalled by the stubs. Storage allocated by other allocators is not released automatically. The manager code must free storage allocated by other allocators. When the manager code makes a remote call, the default memory management routines are **rpc\_ss\_allocate** and **rpc\_ss\_free**.

The syntax of the `rpc_ss_allocate` routine is as follows:

```
idl_void_p_t rpc_ss_allocate (idl_size_t size);
```

The *size* parameter specifies the size of the memory allocated.

**Note:** In American National Standards Institute (ANSI) standard C environments, **idl\_void\_p\_t** is defined as **void \*** and in other environments is defined as **char \***.

Use **rpc\_ss\_free** to release storage that is allocated by **rpc\_ss\_allocate**. You can use the **rpc\_ss\_free** routine to release storage pointed to by a full pointer in an input parameter. You can also have the freeing of the memory reflected on return to the calling application by specifying the `reflect_deletions` attribute as an operation attribute. See **Developing a Simple RPC Application** *Writing an Interface Definition "Operation Declarations"* in the *IBM DCE for AIX, Version 2.2: Application Development Guide—Core Components* for more information.

The syntax of the routine is as follows:

```
void rpc_ss_free (idl_void_p_t node_to_free);
```



The *node\_to\_free* parameter specifies the location of the memory to be freed.

### Enabling and Disabling Memory Allocation

It may be necessary to call manager routines from different environments (for example, when the application is both a client and a server of the same interface). In this case, the same routine may be called both from server manager code and from client code. The **rpc\_ss\_allocate** routine, when used by the manager code to allocate memory, must be initialized before its first use. The stub performs the initialization automatically. Code, other than stub code, that calls a routine, which in turn calls **rpc\_ss\_allocate**, first calls the **rpc\_ss\_enable\_allocate** routine.

The syntax of the routine is:

```
void rpc_ss_enable_allocate (void);
```

The environment set up by the **rpc\_ss\_enable\_allocate** routine is released by calling the **rpc\_ss\_disable\_allocate** routine. This routine releases all memory that was allocated by calls to **rpc\_ss\_allocate** because the call to **rpc\_ss\_enable\_allocate** was made. It also releases memory that was used by the memory management mechanism for internal bookkeeping.

The syntax of the routine is:

```
void rpc_ss_disable_allocate (void);
```



---

## Appendix B. Documentation Provided with this Release

DCE for AS/400 is based on DCE 2.2 for AIX. The documents are available to support a wide range of administrative and application development tasks. These books describe some components and functions that are supported for AIX but not for AS/400.

The following components and functions of DCE 2.2 for AIX are **not** included in this product:

- **Distributed File Service (DFS)** — joins the local file systems of several file server machines, making the file systems equally available to all DFS client machines.
- **CDS, Global Directory Service (GDS), and Security servers** — GDS is a distributed, replicated directory service based on the CCITT X.500/ISO 9594 international standard. It provides a global namespace that connects the local DCE cells into one worldwide hierarchy. CDS (Cell Directory Service) stores the names and the attributes of resources that are located in a DCE cell.
- **Slim Client** — When you configure the DCE software on a client system, more DCE daemons than might be necessary are started. If a client does not offer DCE services to other systems in the cell, it might not need all of the functions provided by these daemons. The *slim client option* avoids starting unnecessary DCE daemons.
- **Integrated Logon** — indicates a logon of DCE with AS/400 user profiles.
- **Audit** — detects and records the running of DCE server operations that are relevant to the maintenance of a secure distributed computing environment.
- **Event Management Service (EMS)** — provides asynchronous event support for DCE-based applications.
- **Simple Network Management Protocol (SNMP)** — provides network management support in the TCP/IP environment for monitoring DCE resources and services.
- **Cell Administration with dcecp**— DCE 2.2 for AS/400 supports only a limited set of dcecp commands for configuration purposes. See “Supported DCECP Commands” on page 35 for a listing of these commands.
- **SMIT** — is an interface specific to AIX that is used to install and configure the DCE product.
- **Public Key Support** — with this functionality, the security server does not need to store the long term key (or password) for a principal. The long term key will remain undisclosed if any compromise of the security server occurs.

- **User-to-User Authentication** — provides an alternate Ticket Granting Service (TGS) protocol. It performs an authenticated Remote Procedure Call (RPC) to a (user-to-user) server on behalf of the client.
- **Global Groups** — allows you to add principals from a foreign cell to a group in the local cell.

This release provides the following DCE 2.2 for AIX information:

- *IBM DCE for AIX, Version 2.2: Introduction to DCE* — Provides an introduction to the IBM Distributed Computing Environment (DCE) offering. The glossary introduces terms that are used in DCE documentation.
- *IBM DCE for AIX, Version 2.2: Administration Guide—Core Components* — Assists system and network administrators in planning, configuring, and managing DCE.
- *IBM DCE for AIX, Version 2.2: Administration Commands Reference* — Assists system and network administrators in using the correct syntax for DCE administration commands. It is divided into technology component sections.
- *IBM DCE for AIX, Version 2.2: Application Development Guide—Introduction and Style Guide* — Provides information about programming DCE in general, and how to use its various components and functions.
- *IBM DCE for AIX, Version 2.2: Application Development Guide—Core Components* — Describes how to use the application program interfaces (APIs) for the various DCE components.
- *IBM DCE for AIX, Version 2.2: Application Development Guide—Directory Services* — Assists programmers in developing applications that use DCE. It also describes guidelines for using DCE features and services.
- *IBM DCE for AIX, Version 2.2: Application Development Reference* — Provides reference material for all of the DCE APIs. It also describes the few commands that are needed by the DCE programmer; in particular, it describes those that are used with the RPC component.
- *IBM DCE for AIX, Version 2.2: Problem Determination Guide* — Lists error messages and recovery actions along with administrative tips and general information. It also explains problem prevention, determination, and resolution. The purpose of this guide is to help programmers and administrators to interpret and to act on error messages and status codes when received.

The DCE 2.2 for AIX documents are available as both HTML **.HTM** files and as printable **.PDF** files as indicated in the following table:

Document	File Names
----------	------------

Introduction to DCE	A3U3SMST.HTM A3U3SMST.PDF
Administration Guide Core Components	A3U3QMST.HTM A3U3QMST.PDF
Administration Commands Reference	A3U3UMST.HTM A3U3UMST.PDF
Application Development Guide Introduction and Style	A3U3HMST.HTM A3U3HMST.PDF
Application Development Guide Core Components	A3U3JMST.HTM A3U3JMST.PDF
Application Development Guide Directory Services	A3U3IMST.HTM A3U3IMST.PDF
Application Development Reference	A3U3KMST.HTM A3U3KMST.PDF
Problem Determination Guide	A3U3LMST.HTM A3U3LMST.PDF

---

## Viewing the HTML Files

You can use any Web browser such as Internet Explorer or Netscape Navigator, to access the HTML versions of the IBM DCE 2.2 documentation at the following location: <http://service.software.ibm.com/dssdce>.

---

## Using the PDF Files

PTF SF48548 will install the PDF files. The `/QIBM/ProdData/DCE/dcedoc/pdf/` directory contains the available versions. After you have installed the files on your system, you can view or print the PDF files by using the Adobe Acrobat Reader. If you do not have the Acrobat Reader installed on your system, you can obtain a free copy at the following location:

<http://www.adobe.com/prodindex/acrobat/readstep.html>



---

## Appendix C. DCE Base Services for AS/400 Application Development Kit

The DCE Base Services for AS/400 (DCE for AS/400) Application Development Kit (ADK) is a development toolkit for a Windows NT workstation. This ADK contains an IDL Compiler, SAMS, DCE Header files, and example programs. You can use the ADK, with the Visual Age Cross Compiler for Windows NT and IBM AS/400 Client Access for Windows NT, to develop DCE applications on a Windows NT workstation for the AS/400.

---

### The ADK and DCE for Windows NT

If a DCE for Windows NT ADK (IBM, Digital, Gradient, Dascom, or other) and a DCE for AS/400 ADK are installed on the Windows NT Workstation, there are file name conflicts with the idl compiler and header files. The names of the idl compiler (idl.exe) and some header files are shared among the DCE ADKs installed on the Windows NT platform.

To avoid problems, do one or more of the following:

- Specify the full path to the correct idl compiler in the make files. For example:

```
Makefile:
    IDL=$(AS4DCEADK)\bin\idl.exe
    ...
    x_cstub.c:
        $(IDL) x.idl...
    x_sstub.c:
        $(IDL) x.idl...
```

**Note:** **AS4DCEADK** is an environment variable that is set at installation that points to the location where the ADK is installed.

- Specify the include path for this ADK as part of your compile flags. For example:

```
Makefile:
    cflags=-I $(AS4DCEADK)\include
        -D IBMAS4NT -D IBMAS4
        -D OS400 -D _MULTI_THREADED

    x.obj
        $(cc) $(cflags) x.c
    ...
    y.obj
        $(cc) $(cflags) y.c
    ...
```

- Ensure that the DCE for AS/400 ADK path locations are the first items in your **include path** and **path** environment variables.

Otherwise you might be using the wrong files because both the DCE for Windows NT ADK and this ADK use the same file names for header files.

---

## Installing the ADK

The installation requirements for the ADK are:

- Visual Age C++ for AS/400
- IBM Client Access for AS/400

The ADK is located on the AS/400 machine that has DCE installed. It is located in **/QIBM/ProdData/DCE/ADK/**.

To install the ADK:

1. With the cursor on the **Network Neighborhood** icon, press the right mouse button and click on **Map Network Drive**.

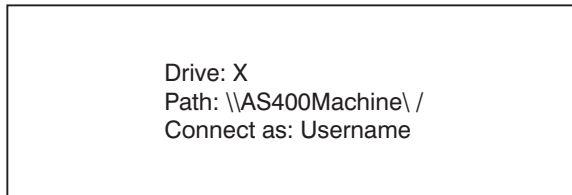


Figure 19. Drive Mount Dialog Box

Type the drive letter you want to mount, the network path and your userid, and then press the ENTER key.

2. On the taskbar, click on **Start**, and then click on **Run**.



Figure 20. Run Dialog Box

3. In the **Run** dialog box, type `X:\QIBM\Proddata\DCE\ADK\setup.exe`, and then press the **ENTER** key.

Where



**X** is the mount drive letter you selected.

4. In the ADK install dialog box  
Select install choices, and then click on **Next**.



Figure 21. ADK Install Dialog Box

5. See the ADK **Readme.txt** located in %AS4DCEADK%\bin for any special considerations.

---

## The Include Path

The include path on the Windows NT development machine plays an important role. It specifies the location where and the order in which the header files are found. The ordering of it is extremely important. The order should be as follows:

```
%AS4DCEADK%\include;%AS4SYSINC%\include;  
%AS4VADRV%:\cttasw\include;%AS4VADRV%:\cttasw\acl\include
```

where

- **AS4DCEADK** is the location of the AS/400 ADK.
- **AS4VADRV** is the drive where the IBM VisualAge C++ is installed.
- **AS4SYSINC** is the location where the native AS/400 system header (included) files are located after being downloaded or copied from the AS/400 systems.

---

## IDL Compiler

The DCE Base Services for AS/400 Interface Definition Language (IDL) compiler generates stub files for remote procedure calls (RPCs).

To list all of the IDL options, on the command line, enter:

```
IDL -confirm
```

The following is an example of an IDL option:

The **-filename** Option

- The **-filename** IDL compiler command option provides backward compatibility support for stubs named with the short filename format.

- The *filename\_format* argument specifies the type of filename format you want to use. If you do not specify this argument, the compiler generates short file names.
- You can specify one of the following values for the *filename\_format* argument:
  - short** Generates stub files with the following format: *file \_c.c* and *file \_s.c*.
  - long** Generates stub files with the following format: *file \_cstub.c* and *file \_sstub.c*.
- The following example command line compiles the IDL interface *test.idl*, and generates stub files that use the short filename format:
 

```
idl test.idl -filename long
```

## Stub Auxiliary Files

By default, IDL compilers in OSF DCE IDL Release 1.1 and later do not generate the *-caux* and *-saux* files. These files would have been generated when an IDL file was compiled with earlier releases. However, if you want to use build procedures that were designed to work with earlier compilers, you can cause the Release 1.1 (and later) IDL compiler to generate empty auxiliary files. To do this, define the environment variable *IDL\_GEN\_AUX\_FILES* with the following command:

```
SET IDL_GEN_AUX_FILES =1
```

See the *DCE Application Development Guide - Core Components* for more comprehensive information and the *IBM DCE Administration Commands Reference* for command information.

---

## SAMS

You use the DCE symbols and message strings (*sams*) utility to generate message catalogs and other files that are used by DCE messaging and servicability APIs. The message catalogs must be installed on the AS/400 in order for the DCE messaging library routines to be able to find them and, consequently, the application messages at runtime. See the **hello\_svc** example program for more details.

Both of the DCE message facilities use XPG4 message catalogs. The *sams* utility generates these message catalogs. You must first define the messages themselves in a text file. The *sams* utility then processes the text files and generates the message catalog file along with other C source files. These files contain the code necessary to facilitate the additional layer of functionality that DCE has added to the XPG4 message catalog mechanism.

The **sams** command will generate files for IBM DCE for Windows NT 2.0 or for DCE for AS/400. To generate files for DCE for AS/400 you must use this command line option:

```
sams -b ibmas4xxx.sams
```

See the **hello\_svc** makefile for an example. The comment at the top of a generated file will indicate that it is an AS/400 application file.

See the *DCE Application Development Guide - Core Components* for more comprehensive information and the *IBM DCE Administration Commands Reference* for command information.

---

## Example Programs

The DCE Runtime Services for the AS/400 ADK supply several example programs. You can locate these programs in directories under **%AS4DCEADK%\examples**. In addition to the information that is provided here, each example program includes an online README file that is located in the same directory as the program. The following table shows the different features for each example program.

Table 11. Example Programs

Example Program	Description
Context_App	A simple client-server application that exercises context handles
RPC Test 1	A simple client-server application that exercises RPCs
RPC Test 2	A simple client-server application that exercises additional RPCs
hello_svc	A simple program to demonstrate messaging and serviceability

You should copy the example program files to a private area before you attempt to build them.

To build the example programs, do the following:

1. Use the **File Manager** or a command such as **xcopy** to copy the files in **%AS4DCEADK%\examples\program\_directory\\*** to your own directory. For example:

```
C:\> cd \mydir
C:\MYDIR> xcopy /s %AS4DCEADK%\examples\rpc\test1
```

2. Use the provided makefile to build the test program . For example:

```
C:\MYDIR> nmake -f test1.mak
```

3. Before building `hello_svc`, you must do some AS/400 setup and make some changes to the makefile. See the makefile readme for detailed instructions.

---

## Generating Universal Unique Identifiers (UUID)

The DCE Base Services for AS/400 ADK does not currently provide a UUID generator program. However, there are several mechanisms for generating UUIDs.

- Use the DCE for AS/400 `uuidgen` program on the AS/400 machine that is shipped with DCE. This is the recommended mechanism.
- Use `uuidgen.exe` that is shipped with IBM DCE for Windows NT.
- Use Microsoft's `uuidgen` program that is shipped with Microsoft's Visual C++.
- Use a `uuid` generator that is provided by various development tools.

---

## Required Compiler Flags

You must define each of the following when compiling DCE for AS/400 applications. Compile with the `-Dx` flag, where `x` is one of the following:

```
IBMAS4NT
IBMAS4
OS400
_MULTI_THREADED
```

Do **not** define the following when compiling DCE for AS/400 applications:

```
WIN32
Intel80x86
DCETHREADS
_ILEC400_
```

---

## Information about the VisualAge Compiler

The VisualAge for C++ for AS/400 is a C++ compiler and not a C compiler. It therefore requires strict ANSI C++ code. You must declare all standard C files and C header files as `extern "C"`. For example:

```
#ifdef _cplusplus
    extern "C" {
#endif

    // C code
```

```
// ...  
#ifdef _cplusplus  
}  
#endif
```

An indication that the C files have not been declared is that linking will fail. Programs will not be created although modules were successfully created. Repeating the link command on the AS/400 (**CRTPGM** for example), with full detail (PF10) shows a failure to find a function name., The function name is mangled. (Mangled is a C++ term for tagging on additional information to a function name when compiling objects.)



---

## Appendix D. Support Services

The IBM DCE Base Services for AS/400 and the IBM DCE Base Services for AS/400 Application Development Kit (ADK) provide the following types of support services.

---

### IBM DCE Base Services for AS/400

The IBM DCE Base Services for AS/400 (5769-DC1) Support Services are provided under the usual terms and conditions for AS/400 software products. Support Services include program services, voice support, and consulting services. Point your web browser to <http://www.as400.ibm.com/> or contact your local IBM representative for more information .

Program services or voice support or both support resolving Distributed Computing Environment (DCE) for AS/400 program defects while consulting services support resolving application programming and debugging issues.

Consulting services support DCE for AS/400 application program interface (API) calls unless:

- It is clearly a DCE API defect as demonstrated by recreation in a relatively simple program.
- It is a question that asks for documentation clarification.
- It is a question on the location of samples or documentation or both.

Consulting services support all programming assistance as well as those program samples that are provided in the IBM DCE Base Services for AS/400 Licensed Program Product. Additional samples may be made available on the Internet at <http://www.as400.ibm.com> on an unsupported basis.

The IBM DCE Base Services for AS/400 Licensed Program Product provides problem solving information. If you believe that there is a potential defect in the DCE for AS/400 API, a simple program that demonstrates the error will be required.





---

## Appendix E. Restrictions and Known Problems

- DCE functions that are expecting strings, should not be passed NULL pointers.
- The **kinit** and **kdestroy** commands have minimal parameter checking. Passing parameters that are not valid may result in unexpected results.
- DCE security API's which are privileged operations must be run by the QDCE user. This includes **sec\_login\_certify\_identity**, and **sec\_login\_valid\_and\_cert\_ident**.
- Use an exit program to defining environment variables (such as NLSPATH) for use with the client access configuration. Use the **WRKREGINF** command to assign to the exit point, **QIBM\_QZRC\_RMT**, a CL program, similar to the following example.

```
/******  
/*  
/* AS/400 SERVERS- SAMPLE USER EXIT PROGRAM          */  
/*  
/* THE FOLLOWING CONTROL LANGUAGE PROGRAM UNCONDITIONALLY  
/* ACCEPTS ALL REQUESTS. IT CAN BE USED AS A SHELL FOR DEVELOPING */  
/* EXIT PROGRAMS TAILORED FOR YOUR OPERATING ENVIRONMENT.      */  
/*  
/*  
/******  
PGM PARM(&STATUS &REQUEST)  
  
/* * * * * * * * * * * * * * * * * * * * * * * * * */  
/*  
/* PROGRAM CALL PARAMETER DECLARATIONS */  
/*  
/* * * * * * * * * * * * * * * * * * * * * * * * * */  
  
/* Return a '1' to accept, '0' to reject */  
DCL VAR(&STATUS) TYPE(*CHAR) LEN(1) /* Accept/Reject indicator */  
/* Note: Request is declared as *CHAR LEN(2000) because that is */  
/* the limit in CL. The actual length of REQUEST is 4171. */  
/* */  
DCL VAR(&REQUEST) TYPE(*CHAR) LEN(2000) /* Parameter structure */  
  
/******  
/*  
/* PARAMETER DECLARES          */  
/*  
/******  
  
/* COMMON DECLARES */  
DCL VAR(&USER) TYPE(*CHAR) LEN(10)  
/* User ID      */  
DCL VAR(&APPLIC) TYPE(*CHAR) LEN(10)  
/* Server ID   */
```

```

DCL VAR(&CSFMT) TYPE(*CHAR) LEN(8) /* Format of function */

/* OPTIMIZED REMOTE COMMAND SERVER DECLARES */
DCL VAR(&RCFMT) TYPE(*CHAR) LEN(8) /* Format name */
DCL VAR(&RCFID) TYPE(*CHAR) LEN(4) /* Function identifier */
DCL VAR(&RCPGM) TYPE(*CHAR) LEN(10) /* Program name */
DCL VAR(&RCLIB) TYPE(*CHAR) LEN(10) /* Program library name */
DCL VAR(&RCNUM) TYPE(*CHAR) LEN(4) /* Number of parms or cmdlen */
DCL VAR(&RCDATA) TYPE(*CHAR) LEN(30)/* Command string parms */

/*****/
/* */
/* OTHER DECLARES */
/* */
/*****/
DCL VAR(&WRKLEN) TYPE(*CHAR) LEN(5)
DCL VAR(&DECLEN) TYPE(*DEC) LEN(8 0)
DCL VAR(&STRING) TYPE(*CHAR) LEN(70)

MONMSG CPF0000 EXEC(GOTO CMDLBL(EXIT))
/* * * * * * */
/* */
/* EXTRACT THE VARIOUS PARAMETERS FROM THE STRUCTURE */
/* */
/* * * * * * */

CHGVAR VAR(&STATUS) VALUE('1') /* INITIALIZE RETURN +
                               VALUE TO ACCEPT THE REQUEST */

/* HEADER */
CHGVAR VAR(&USER) VALUE(%SST(&REQUEST 1 10))
CHGVAR VAR(&APPLIC) VALUE(%SST(&REQUEST 11 10))
CHGVAR VAR(&CSFMT) VALUE(%SST(&REQUEST 21 8))

IF COND(&CSFMT *EQ 'CZRC0100') THEN(DO)
CHGVAR VAR(&RCFMT) VALUE(%SST(&REQUEST 21 8))
CHGVAR VAR(&RCFID) VALUE(%SST(&REQUEST 29 4))
CHGVAR VAR(&RCPGM) VALUE(%SST(&REQUEST 33 10))
CHGVAR VAR(&RCLIB) VALUE(%SST(&REQUEST 43 10))
CHGVAR VAR(&RCNUM) VALUE(%SST(&REQUEST 53 4))
CHGVAR VAR(&RCDATA) VALUE(%SST(&REQUEST 57 30))
ENDDO

/*****/
/* */
/* BEGIN MAIN PROGRAM */
/* */

/* ADD LOGIC COMMON TO ALL SERVERS */

/* PROCESS BASED ON SERVER ID */
IF COND(&APPLIC *EQ '*RMTSRV') THEN(GOTO CMDLBL(RMTCMD)) /* IF RMTCMD/DPC */

```

```

GOTO EXIT

/* * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* SUBROUTINES                                          */
/*                                                    */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * */

/* REMOTE COMMAND/DISTRIBUTED PROGRAM CALL */
RMTCMD:
  IF COND(&USER *EQ 'QDCE      ') THEN(DO)
    ADDENVVAR ENVVAR(NLSPATH) +
      VALUE('/QIBM/ProdData/OS400/Shell/MRI2924/%N:
            /QIBM/ProdData/DCE/usr/lib/nls/msg/MRI2924/%N')
    MONMSG CPF0000
  ENDDO
  GOTO EXIT

EXIT:

/* Accept everything */
CHGVAR VAR(&STATUS) VALUE('1') /* RETURN +
                                VALUE TO ACCEPT THE REQUEST */
ENDPGM

```

---

## National Language Support (NLS) Restrictions

- DCE does not run on Asian installations of AS/400.
- DCE does not, and will never, support the Japanese SBCS CCSID 290.
- AS/400 DCE does not support the IBM PGO naming extensions that are enabled by DCE\_USE\_NONPORTABLE\_NAMES. Do not use these extensions in any DCE enterprise that includes AS/400 DCE.
- The RPC csrc compiler must be run in coded character set identifier (CCSID) 037. In general, it is not necessary and not recommended that DCE users run this compiler.
- **dcecp** is the standard control program for DCE, and it has been designed to support a variety of country environments. As noted in the DCE documentation, dcecp replaces several older control programs (**cdscp**, **dtscp**, **rpccp**, **acl\_edit**, **rgy\_edit**, and **sec\_admin**). These older programs were not designed for international use, and may give unexpected or undesirable results when used in non-English environments.
- While **dcecp** supports non-English data, there are some restrictions. **dcecp** string handling commands, such as string range, have byte-based, not character-based, semantics. They may give undesired results when used on characters outside of the DCE Portable Character Set.

---

## Message Handling Restrictions and Serviceability Restrictions

- **dce\_svc\_debug\_set\_levels()** does not work correctly and should not be used. Set the debug levels in the routing string instead.
- **dce\_svc\_printf()** should be used only for production messages, not for debug messages. Use **DCE\_SVC\_LOG** or **DCE\_SVC\_DEBUG** instead.
- Binary serviceability logs contain encoded log records which must be decoded by the function **dce\_svc\_log\_get()** before they can be viewed. Occasionally, a damaged record will be written to the log, or a user will inadvertently mix text and binary log records in the same log file. When the data that is not valid is read by **svcdumplog** or some other application which calls **dce\_svc\_log\_get()**, error reporting may be cryptic. To recover, remove the log file that contains the bad data. Serviceability will create a new log file.
- Binary serviceability logs use message catalogs to resolve the message text. If message catalogs are not available, the variant EBCDIC characters in messages may not display correctly.

---

## Appendix F. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, NY 10594  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-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 publication. IBM may make

improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licenses 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  
Software Interoperability Coordinator  
3605 Highway 52 N  
Rochester, MN 55901-7829  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

---

## Trademarks

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX  
AS/400  
Client Access  
Client Access/400  
IBM  
MVS  
Operating System/2  
OS/2  
VisualAge

DFS is trademark of Transarc Corporation

Microsoft, Windows, Windows NT®, and the Windows logo are registered trademarks of Microsoft Corporation.

Open Software Foundation, OSF, the OSF logo, OSF/1, OSF/Motif, and Motif are registered trademarks of the Open Software Foundation, Inc.

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





---

# Index

## A

- Adobe Acrobat Reader
  - how to obtain 57
- Application Development Kit (ADK)
  - conflicting file names 59
  - example programs
    - Context\_App 63
    - hello\_svc 63
    - RPC Test 1 63
    - RPC Test 2 63
  - IDL Compiler Enhancements 61
    - filename 61
    - stub auxiliary files 62
  - installation 60
  - SAMS 62
- automatically starting DCE at boot
  - time 16

## C

- CCSID 39
- CCSID conversion tables 39
- CDMF 34
- cell
  - definition 1
- cell directory client 33
- cells 11
  - joining an existing 11
- checking your configuration 24
- client code 48
- client programs
  - CDS client
    - cdsadv 5
    - cdsclerk 5
  - DTS 6
  - RPC 5
  - security service 5
- commands
  - supported
    - configuration 34
    - DCE 34
    - dcecp 34
- common data masking facility 34
- compiler
  - defining flags 64
  - VisualAge 64
- configuration commands 34
- configuring
  - checking 24
  - correcting errors 28

- configuring (*continued*)
  - deleting 22
  - full client 15
  - local client 19
  - modifying 21
  - to use DTS 17
- configuring DCE 14
- configuring DCE for AS/400 11

## D

- DCE
  - automatically starting at boot
    - time 16
  - cells 11
  - configuring 11
  - description 1
  - documentation provided with
    - 56
  - installing 9
  - running configuration 12
  - supported AS/400 functions 33
  - unsupported functions 55
- DCE commands 34
- dcecp
  - supported commands 35
- defining a hostname 11
- deleting your configuration 22
- distributed time service 6
- documentation 55
  - online 57
- documentation file names 56
- documentation provided 56
- DTS
  - planning 6
- DTS, configuring 17
- dtstd 6

## E

- EBCDIC 39
- environment variables
  - NLSPATH 37
- example program
  - hello\_svc 42

## F

- freeing memory
  - dce\_free 46
- full client, configuring as 15
- full client configuration 14

## G

- GSSAPI extensions 33

## I

- IDL 50
- IDL compiler 34
- interface definition language 50

## L

- local client, configuring as 19
- local client configuration 14
- locale 9
  - setting
    - POSIX 42
    - XPG4 42
- log files
  - last configuration 26
  - size 27
  - viewing
    - full configuration 26

## M

- manager code
  - routines
    - rpc\_ss\_allocate 48
    - rpc\_ss\_free 48
- memory
  - allocation 46
  - freeing 46
- message handling 37
- migration 9
- modifying your configuration 21
- multithreaded programming
  - environment 33

## N

- national language support
  - CCSID 37
  - POSIX 37
  - setting locale 9
  - XPG4 37
- notices 73

## O

- Open Software Foundation 1

## P

- product components 5

## Q

- QDCE user profile 9

QSHELL  
  commands  
    config.dce 18

## R

Remote Procedure Call 5  
required software 3  
restrictions  
  environment variables  
    exit program 69  
  kdestroy command 69  
  kinit command 69  
  message handling  
    debug 72  
  national language support  
    asian installations 71  
    dcecp 71  
    Japanese SBCS CCSID 290 71  
    PGO naming extensions 71  
    RPC csrc compiler 71  
  NULL pointers 69  
  security APIs 69  
  serviceability  
    binary logs 72  
RPC 5  
RPC client stubs 47

## S

security service 5  
server processes  
  DTS 6  
services  
  dtsd 6  
software requirements 3  
support services 67  
  DCE Base Services 67  
supported dcecp commands 35  
supported functions 33  
system requirements  
  software 3

## T

technology components  
  Distributed Time Service 33  
  multithreaded programming  
    environment 33  
  RPC 33  
threads  
  draft levels 45  
  header files 45  
  single threaded 33  
trademarks 75  
types of modifications 21

## U

unconfiguration  
  admin 9  
  admin unconfiguration 22  
  delete 22  
  unconfig.dce command 9  
unsupported DCE functions 55  
UUID  
  generation of 64

## V

viewing and correcting errors 28

## X

XPG4 37

---

## Readers' Comments — We'd Like to Hear from You

AS/400e series

IBM Distributed Computing Environment Version 2.2, Base Services for AS/400: Getting Started

Publication No. SC41-5601-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

\_\_\_\_\_

Name

\_\_\_\_\_

Address

\_\_\_\_\_

Company or Organization

\_\_\_\_\_

Phone No.



Cut or Fold  
Along Line

Fold and Tape

Please do not staple

Fold and Tape



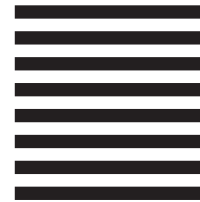
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation  
ATTN DEPT 542 IDCLERK  
3605 HWY 52 N  
ROCHESTER MN  
55901-7829



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold  
Along Line





Part Number: 24L2114



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SC41-5601-00



24L2114



Spine information:



AS/400e series

Getting Started

SC41-  
5601-00