

Intel® NetMerge™ Call Processing Software

Installation and Configuration Guide

Order Number: 05-0990-005

Software/Version: Intel NetMerge Call Processing Server Version 6.0
Intel NetMerge Call Processing Application Programming
Interface Version 6.0

Server Platform: Windows NT†/Windows† 2000/Windows XP

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

This document as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without express written consent of Intel Corporation.

Copyright © 2002 Intel Corporation.

Intel and NetMerge are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Acrobat and Adobe are trademarks of Adobe Systems, Incorporated.

†Other names and brands may be claimed as the property of others.

Publication date: September, 2002

For **Sales Offices** and other contact information, visit our web site at <http://www.intel.com>.

Contents

About This Manual	xiii
--------------------------------	------

1 Installation Overview

1.1	Installation Procedure	1-1
1.2	Software Overview	1-2
1.2.1	Server Software	1-2
1.2.2	API Software	1-2
1.2.2.1	Client Platforms Supported	1-3
1.2.2.2	Server Interfaces to Clients	1-3
1.2.2.3	Running Applications on the Server	1-3
1.2.2.4	Java Client Communication Through a Firewall	1-3
1.2.3	Management API Software	1-4
1.3	Connecting Clients and Servers Across Firewalls	1-4
1.4	Upgrading From CT Connect	1-4

Part I Installing and Configuring the Call Processing Server Software

2 Installing the Call Processing Server Software

2.1	Overview	2-1
2.2	Kit Contents	2-1
2.3	Hardware Requirements	2-2
2.4	Software Requirements	2-3
2.4.1	Configuring the DIVA ISDN Driver Software	2-4
2.4.2	Configuring an EiconCard X.25 Network Adapter Card	2-5
2.5	Attaching the Hardware License Key	2-7
2.5.1	Types of Hardware Key	2-7
2.5.2	Attaching the Hardware Key	2-8
2.6	Installation Procedure	2-8
2.7	Files Installed	2-10

2.8	What to Do Next	2-11
2.8.1	Reading the Release Notes	2-11
2.8.2	Reading the Online Documentation	2-12
2.8.3	Reading the Java API Programming Documentation	2-12
2.9	Reinstalling the Call Processing Server Software	2-13
2.10	Removing the Call Processing Server Software.	2-13
2.10.1	Windows 9x and Windows NT	2-13
2.10.2	Windows 2000	2-13
2.10.3	Windows XP	2-13

3 Configuring the Call Processing Server Software

3.1	Introduction.	3-1
3.1.1	What is a Server Communications Link?	3-1
3.1.2	The Configuration Program	3-2
3.1.2.1	Authorization Setup Program	3-2
3.1.2.2	Distributed Data Setup Program	3-2
3.2	Configuring the Server and Creating Links: Procedure	3-2
3.3	Required Configuration Information	3-3
3.4	Advanced Settings	3-15
3.4.1	Link State Checking	3-15
3.4.2	Event Buffering	3-16
3.4.3	Route Point Buffering	3-16
3.4.4	Data Parameters	3-16
3.4.4.1	Swap Call References	3-16
3.4.4.2	Server Application Data	3-16
3.5	Maximum Monitors	3-17
3.6	Configuring Additional Links.	3-17
3.7	Modifying a Link	3-17
3.8	Deleting a Link	3-18
3.9	Leaving the Configuration Program	3-18
3.10	Checking the Communications Link to the Switch	3-19

4 Configuring Device Level Authorization

4.1	Introduction.	4-1
4.1.1	What is Device Level Authorization?	4-1
4.1.2	Configuring Device Level Authorization	4-1
4.1.3	Checking User Authorization	4-2
4.1.4	Upgrading Client Applications for Device Level Authorization	4-3
4.2	Authorization Setup Program: Information Required	4-3

4.2.1	User Name	4-3
4.2.1.1	User Names and Password Checking Options	4-3
4.2.1.2	Login User Name (Password: None)	4-4
4.2.1.3	Domain User Name (Password: System)	4-4
4.2.1.4	Local User Name (Password: System)	4-4
4.2.1.5	Application User Name: (Password: Application).	4-5
4.2.1.6	Client Applications and User Name and Password Checking . . .	4-5
4.2.2	Password	4-5
4.2.3	Devices	4-5
4.2.3.1	Device Type	4-6
4.2.3.2	Device Number	4-6
4.2.3.3	Range.	4-7
4.2.3.4	Link Name	4-7
4.2.3.5	Permission Level	4-7
4.2.4	Client System Names	4-7
4.3	Using the Authorization Setup Program	4-7
4.4	Authorization Setup Program: Deleting and Modifying Information	4-8
4.4.1	Deleting Information	4-8
4.4.2	Modifying Device Information	4-8
4.4.3	Modifying a Password	4-9
4.5	The Import/Export Feature	4-9
4.5.1	How to Import or Export	4-9
4.5.2	Importing and Exporting in the Authorization Setup Program	4-9
4.5.3	Using the Import/Export Utility	4-10
4.5.3.1	Running the Import/Export Utility	4-10
4.5.4	Authorization File Format	4-11
4.5.4.1	Defining Passwords	4-12
4.5.4.2	Multiple Device Numbers in the Authorization File.	4-12
4.5.4.3	Multiple Devices and Client Systems for a Single User	4-12
4.5.4.4	Command Execution	4-12
4.5.4.5	RemoveUser Commands in Exported Authorization Files.	4-12
4.5.5	Example Authorization File.	4-13

5 Configuring Distributed Application Data

5.1	Introduction.	5-1
5.1.1	Background	5-1
5.1.2	Configuring Distributed Data	5-1
5.1.2.1	Basic Configuration Method	5-2
5.1.2.2	Example	5-2
5.1.3	How Data is Distributed	5-3
5.2	Configuring the Distributed Data Network	5-3

5.2.1	Distributed Data Requirements	5-3
5.2.2	Before You Begin	5-4
5.2.3	Required Information	5-4
5.2.4	Starting the Distributed Data Setup Program	5-6
5.2.5	General Configuration Method	5-6
5.3	Example Configuration	5-6
5.3.1	Configure Server 1	5-7
5.3.2	Configure Server 2	5-8
5.3.3	Result of Configuration	5-8
5.4	Modifying Link Information	5-9
5.5	Deleting Link Information	5-9

6 Configuring Web Access Using SOAP

6.1	Introduction	6-1
6.1.1	What is SOAP?	6-1
6.1.2	Using the Jakarta Tomcat Servlet Container	6-1
6.1.3	Call Processing Software SOAP Configurations	6-2
6.1.4	Overview: Java Client Access	6-2
6.1.5	Overview: Distributed Data Servers	6-3
6.2	Installation	6-4
6.2.1	Product Versions Installed	6-5
6.3	Integrating Tomcat With Your Web Server	6-5
6.3.1	Integrating Tomcat with an Apache Web Server	6-5
6.3.2	Integrating Tomcat with an IIS Web Server	6-6
6.4	Configuring SOAP to Use Log Files	6-6
6.4.1	Logging Call Processing API Messages	6-6
6.4.2	Logging Distributed Data Messages	6-7
6.4.3	Location of Log Files	6-7
6.5	Using Soap on an External Java Client	6-7
6.5.1	Installing the Required Software	6-7
6.5.2	Programming Arguments	6-7
6.6	Using Soap on a Distributed Data Server	6-8
6.7	Supported Methods for Java Clients	6-8

7 Starting and Stopping the Call Processing Server

7.1	Starting the Server Software	7-1
7.1.1	Automatic Startup	7-1
7.1.2	Starting the Server Software Manually	7-2
7.1.2.1	Windows NT	7-2

7.1.2.2	Windows 2000	7-2
7.1.2.3	Windows XP	7-2
7.1.2.4	Events Logged	7-2
7.2	Stopping the Server Software	7-3
7.2.1	Windows NT	7-3
7.2.2	Windows 2000	7-3
7.2.3	Windows XP	7-3
7.2.4	Events Logged	7-4
7.3	Starting a Link Between the Server and the Switch	7-4
7.3.1	Starting the Link Automatically	7-4
7.3.2	Starting the Link Manually	7-4
7.3.3	Events Logged	7-4
7.3.4	Errors at Link Startup	7-4

Part II Installing the Call Processing API Software

8 Installing the Call Processing API on HP-UX

8.1	Overview	8-1
8.2	Hardware Requirements	8-1
8.3	Software Requirements	8-2
8.3.1	Defining the Call Processing Server Name and Address	8-2
8.4	Installation Procedure	8-2
8.5	Files Installed	8-3
8.6	What to Do Next	8-4
8.7	Checking Communication	8-4
8.7.1	Procedure	8-4
8.7.2	Using CTC Test with Device Level Authorization	8-5
8.7.2.1	Procedure	8-5
8.7.3	What to Do If CTC Test Fails	8-6
8.8	Removing the Software	8-6

9 Installing the Call Processing API on OpenVMS

9.1	Overview	9-1
9.2	Hardware Requirements	9-1
9.3	Software Requirements	9-2
9.3.1	DCE RPC Environment	9-2
9.3.2	Privileges	9-2
9.3.3	Defining the Call Processing Server Name and Address	9-3
9.4	Installation Procedure	9-3

9.4.1	The IVP Fails	9-5
9.5	Files Installed	9-5
9.6	What to Do Next	9-6
9.6.1	Starting the Software	9-6
9.7	Checking Communication.	9-6
9.7.1	Procedure.	9-6
9.7.2	Using CTC Test with Device Level Authorization	9-7
9.7.2.1	Procedure.	9-7
9.7.3	What to Do If CTC Test Fails	9-8

10 Installing the Call Processing API on Solaris

10.1	Overview.	10-1
10.2	Hardware Requirements.	10-1
10.3	Software Requirements	10-2
10.3.1	Defining the Call Processing Server Name and Address	10-2
10.4	Installation Procedure.	10-2
10.5	Files Installed	10-3
10.6	What to Do Next	10-4
10.7	Checking Communication.	10-4
10.7.1	Procedure.	10-4
10.7.2	Using CTC Test with Device Level Authorization	10-5
10.7.2.1	Procedure.	10-6
10.7.3	What to Do If CTC Test Fails	10-6
10.8	Removing the Software	10-7
10.9	Reinstalling the Call Processing API Software.	10-7

11 Installing the Call Processing API on Tru64 UNIX

11.1	Overview.	11-1
11.2	Hardware Requirements.	11-1
11.3	Software Requirements	11-2
11.3.1	Defining the Call Processing Server Name and Address	11-2
11.4	Installation Procedure.	11-2
11.4.1	The IVP Fails	11-3
11.5	Files Installed	11-4
11.6	What to Do Next	11-5
11.7	Checking Communication.	11-5
11.7.1	Procedure.	11-6
11.7.2	Using CTC Test with Device Level Authorization	11-6
11.7.2.1	Procedure.	11-7

11.7.3	What to Do If CTC Test Fails	11-7
11.8	Removing the Software	11-8

12 Installing the Call Processing API on Windows Systems

12.1	Overview	12-1
12.2	Hardware Requirements	12-1
12.3	Software Requirements	12-2
12.3.1	Windows 9x	12-2
12.3.2	Windows NT	12-2
12.3.3	Windows 2000	12-2
12.3.4	Windows XP	12-2
12.3.5	All Windows Systems	12-2
12.4	Other Installation Requirements	12-3
12.4.1	Defining the Call Processing Server Name and Address	12-3
12.5	Installation Procedure	12-3
12.6	Files Installed	12-4
12.7	What to Do Next	12-4
12.8	Checking Communication	12-4
12.8.1	Procedure	12-5
12.8.2	Using CTC Test with Device Level Authorization	12-6
12.8.2.1	Procedure	12-6
12.8.3	What to Do If CTC Test Fails	12-6
12.9	Removing the Software	12-7
12.9.1	Windows 9x and Windows NT	12-7
12.9.2	Windows 2000	12-7
12.9.3	Windows XP	12-7
12.10	Reinstalling the Call Processing API	12-8

13 Installing the Call Processing API for Java

13.1	Overview	13-1
13.2	Hardware Requirements	13-1
13.3	Software Requirements	13-2
13.3.1	Defining the Call Processing Server Name and Address	13-2
13.4	Installation Procedure	13-2
13.4.1	Required Files	13-3
13.4.2	Method 1	13-4
13.4.3	Method 2	13-5
13.4.3.1	Before You Begin	13-5
13.4.3.2	Procedure	13-6

13.5	Files Installed	13-6
13.6	What to Do Next	13-7
13.7	Checking Communication.	13-7
13.7.1	Procedure	13-7
13.7.2	Using CTC Test with Device Level Authorization	13-8
13.7.2.1	Procedure.	13-8
13.7.3	Using CTC Test On External Java Clients	13-9
13.7.3.1	Procedure.	13-9
13.7.4	What to Do If CTC Test Fails	13-9
13.8	Removing the Software	13-10

14 What to Do Next

Index

Figures

2-1	Call Processing Software Hardware Key	2-7
4-1	Checking User Authorization	4-2
5-1	Example Distributed Data Network.	5-2
5-2	Distributed Data Configuration	5-7
5-3	Link Pairs for Server 1 and Server 2	5-9
6-1	Client Access Through an IP Firewall.	6-2
6-2	Distributed Data Access Through an IP Firewall	6-3

Tables

2-1	Supported Network Cards	2-2
2-2	Supported Network Card Software.	2-3
2-5	Files Installed on the Call Processing Server.	2-10
3-1	Configuration Information for All Switches	3-4
3-2	Configuration Information for CSTA Switches	3-9
3-3	Configuration Information for Avaya DEFINITY Switches	3-12
3-4	Configuration Information for Meridian 1 Switches.	3-13
4-1	Device Types	4-6
4-2	Device Numbers.	4-6
4-3	Authorization File Keyword Commands	4-11
5-1	Distributed Data Setup Information Required.	5-4
6-1	SOAP Tasks.	6-4
6-2	Software Installed to Implement SOAP	6-5
8-1	Call Processing API Files Installed on HP-UX	8-3

9-1	Call Processing API Files Installed on OpenVMS	9-5
10-1	Call Processing API Files Installed on Solaris	10-4
11-1	Call Processing API Files Installed on Tru64 UNIX	11-4
11-2	Tru64 UNIX Symbolic Links	11-5
12-1	Call Processing API Files Installed on Windows Systems	12-4
13-1	Java Client API Files and Supporting Files	13-3
13-2	External Access Client Files	13-4
13-3	Documentation Files	13-4

About This Manual

This manual describes how to install and configure the Intel NetMerge Call Processing Software.

It contains the following:

- Chapter 1 provides an overview of the information in this guide. It summarizes the installation and configuration procedures and the tasks that you will need to perform.
- Part I describes how to install the call processing server software and configure links to the switch:
 - Chapter 2 contains instructions for installing the server software on a system running Microsoft† Windows NT, Windows 2000 or Windows XP.
 - Chapter 3 describes how you configure the links between the server and your switch and how you check that the server and the switch can communicate.
 - Chapter 4 describes how to configure device level authorization on the call processing server.
 - Chapter 5 describes how to configure the distributed data feature on the call processing server.
 - Chapter 6 describes how the Call Processing Software uses the Simple Object Access Protocol (SOAP) to pass messages through a firewall, and what you need to do to use this feature.
 - Chapter 7 describes how to start and stop the call processing server software.
- Part II describes how to install the call processing client software:
 - Chapters 8 to 13 contain instructions for installing the Application Programming Interface (API) software on the supported operating system(s) and environments.

Audience

This manual is for anyone responsible for installing and configuring the Call Processing Software.

To follow the procedures, you must understand what the software is and how it works. You will find the necessary information in the *Intel NetMerge Call Processing Software Introduction*.

You will also need:

- Experience of installing and managing software on the appropriate operating systems.
- Privileges for installing software on the appropriate operating systems. For example, for installing the API software on a Windows NT system, administrator privileges.
- Access to information about the configuration of your switch and its requirements. For example, you may need to contact your switch administrator for details of specific configuration options set on your switch.

Associated Documentation

Intel NetMerge Call Processing Software Documentation

In addition to this manual, the following are included in the Intel NetMerge Call Processing Software documentation set:

- *Intel NetMerge Call Processing Software Introduction*—This manual provides an overview of the Call Processing Software and example network configurations. It is provided in hard copy and as a Portable Document Format (PDF) file.
- *Intel NetMerge Call Processing Software C Programming Guide*—This manual provides detailed descriptions of the API procedural routines and guidelines for using them. It also includes details of the operational differences for links to specific switches. This manual is provided in hard copy and as a PDF file.
- *Intel NetMerge Call Processing API for the Java™ Platform*—This information provides a detailed description of the API for the Java™ platform. This defines the Call Processing Software packages, interfaces, and classes, along with the fields and methods supported. It also includes details of the operational differences for links to specific switches. This information is provided as a series of HTML files.

- *Intel NetMerge Call Processing Software Management API C Programming Guide*—This manual describes all management API routines, and provides guidelines for creating a management application similar to the Control Program. This manual is provided only as a PDF file.
- *Intel NetMerge Call Processing Management API for the Java™ Platform*—This information provides a detailed description of the management API for the Java platform. This defines the Call Processing Software packages, interfaces, and classes, along with the fields and methods supported. This information is provided as a series of HTML files.
- *Intel NetMerge Call Processing Software CTC Test User's Guide*—This manual describes how to use the Call Processing Software application, CTC Test. You can use CTC Test to check the validity of function sequences you expect to make using your call processing application. This manual is provided as a PDF file.
- *Intel NetMerge Call Processing Software Release Notes*—These online notes provide information about changes to the software and/or documentation at the time of release. This document is provided as a text file.

All online files, and the Adobe™ Acrobat™ Reader you use to view PDF files, can be installed at the same time as the call processing server software. For details, refer to Chapter 2.

Intel Web Site

For more information about Call Processing Software, and other Intel products, visit Intel's web site at <http://www.intel.com>.

Switch Documentation

Refer to the documentation supplied with the switch for details of features, and any limitations that may affect the operation of the Call Processing Software.

Terms and Definitions

The following terms are used throughout this manual:

Term	Definition
Call processing API for the Java platform	The Intel NetMerge Call Processing Software API for the Java platform.
Call processing API	The Intel NetMerge Call Processing Software API for all supported platforms

Term	Definition
Client or call processing client	A supported system that has the call processing API software installed, and is running a call processing application.
Server or call processing server	A Windows NT, Windows 2000 or Windows XP personal computer running the call processing server software.
Communications link	A logical link between the server and the switch.
Switch	The telephony switching device, for example, a Private Automated Branch Exchange (PABX), Private Branch Exchange (PBX), or a central office switch.
Nortel Meridian 1	Unless otherwise stated, refers to any of the following: Nortel† Meridian 1† with Meridian Link Module, Meridian Link Services, Symposium† Call Center Server, or Symposium Express Call Center.
Avaya DEFINITY	The range of communications systems that support the Avaya† CallVisor† ASAI protocol over a LAN Gateway or Multi Application Platform for DEFINITY† (MAPD).
OpenVMS	Refers to the OpenVMS† Alpha operating system.
Tru64 UNIX	Refers to the Compaq† Tru64† UNIX† operating system.
Windows 9x	Unless otherwise stated, refers to the Windows 95, Windows 98 and Windows Me operating systems.
Programs on the Start Menu	Refers either to the Programs item on the Start menu on Window 9x, Windows NT and Windows 2000 systems, or to the All Programs item on the Start menu on Window XP systems.
<i>drive:\directory</i>	The installation directory. This is C:\Program Files\Intel\ctc by default.
<i>drive:\</i>	The installation drive. This is C:\ by default.

Conventions

The following table shows the conventions used in this manual:

Convention	Meaning
<i>courier</i>	This typeface is used for code examples or interactive examples to indicate system input/output.
<i>drive:</i>	Italic (slanted) typeface indicates variable values, placeholders and function arguments.
#	The Tru64 UNIX or HP-UX† command prompt.
\$	The OpenVMS command prompt.

Installation Overview

This chapter provides an overview of the procedures for installing the Intel NetMerge Call Processing Software. The procedures include configuring the software and checking communications between the server, the switch, and the client systems.

1.1 Installation Procedure

This section summarizes the procedures for installing, configuring and checking the Call Processing Software. If you are not familiar with the software, refer to Section 1.2 for an overview of both the call processing server software and the call processing API software.

The Call Processing Software was formerly known as CT Connect (CTC). If you are upgrading from CTC, refer to Section 1.4 for information about version compatibilities.

To install and configure your Call Processing Software, follow these steps:

1. Install the call processing server software. Refer to Chapter 2.
2. Configure and check the communication links between the server and the switch. Refer to Chapter 3.
3. Install the call processing API software on your client systems. Refer to the chapter dedicated to your client system, as listed in the following table:

For this type of client...	Follow the procedure in...
Distributed Computing Environment (DCE) Based	
HP-UX	Chapter 8
OpenVMS	Chapter 9
Solaris	Chapter 10
Tru64 UNIX	Chapter 11
Windows, including Windows 9x, Windows NT, Windows 2000, and Windows XP	Chapter 11

For this type of client...	Follow the procedure in...
Java Remote Method Invocation (RMI) Based	
Java Runtime Environment	Chapter 13

4. Proceed to the post-installation steps. Refer to Chapter 14.

1.2 Software Overview

The Call Processing Software comprises:

- Server software
- Application Programming Interface (API) software

1.2.1 Server Software

You install the call processing server software on a Windows NT, Windows 2000 or Windows XP PC system connected to the following:

- A telephony switching device (for example, a PBX) by means of a TCP/IP, X.25, ISDN Basic Rate, or V.24 connection.
- One or more API client systems in the same network as the PC

The server acts as an intermediary, passing messages between the switch and the API client systems.

1.2.2 API Software

There are two programming interfaces you can use to create a telephony application:

- The procedural API for writing an application in C (or other procedural language). This requires Distributed Computing Environment (DCE) based development software.
- The API for the Java™ platform for writing an application in the Java programming language. This requires the Java 2 Runtime Environment (JRE) and, for development, the Java Software Development Kit (SDK).

You install the API software and your call processing application on one or more client systems in the same network as the server. A system running the call processing API software is known as a call processing client.

For more information about the call processing API, refer to these programming guides:

- *Intel NetMerge Call Processing Software C Programming Guide*
- *Intel NetMerge Call Processing API for the Java Platform* (online)

1.2.2.1 Client Platforms Supported

The client platforms supported are as follows:

- Operating Systems
 - HP-UX
 - OpenVMS
 - Solaris
 - Tru64 UNIX
 - Windows (Windows 9x, Windows NT, Windows 2000 and Windows XP)
- Runtime Environment
 - Java 2 Runtime Environment (JRE)

1.2.2.2 Server Interfaces to Clients

The server uses a Java RMI interface to clients running the JRE, and a Distributed Computing Environment/Remote Procedure Call (DCE/RPC) interface to other clients. Both interfaces can be installed as part of the server software. Refer to Chapter 2 for details.

1.2.2.3 Running Applications on the Server

You can run call processing applications on the server by installing both the API software and the server software on the same PC. However, this may affect its performance as a call processing server (depending on resources available).

If you write a Java call processing applications as a Java applet, you can run the applet on the server, and make it available to a Web browser on the Java client. In this case, the Web server needs to be on the same system as the call processing server.

1.2.2.4 Java Client Communication Through a Firewall

The Call Processing Software allows Java call processing applications to communicate with a call processing server through a firewall. Refer to Section 1.3 and Chapter 6 for more information.

1.2.3 Management API Software

When you install the call processing API, a management API is also copied to the client system. You can, if you wish, use the management API to create an application to manage the call processing server. However, note that it is not necessary to do this, as you can if you wish use the Call Processing Software Control Program to manage the call processing software. The Control Program is a management application that is provided as part of the product.

For more information about the management API, refer to the *Intel NetMerge Call Processing Software Management API C Programming Guide* and the *Intel NetMerge Call Processing Management API for the Java Platform*.

1.3 Connecting Clients and Servers Across Firewalls

The Call Processing Software can make use of the Standard Object Access Protocol (SOAP) to allow Call Processing Software messages to travel through firewalls. There are two possible configurations:

- External call processing clients use the Java call processing API to communicate with the call processing server through a firewall.
- Call processing servers exchange distributed data through firewalls. Distributed data is described in Chapter 5.

Refer to Chapter 6 for more information.

1.4 Upgrading From CT Connect

If you are running CTC version 3.0, 4.0 or 5.0 applications on a number of systems, you can upgrade your call processing network progressively, as follows:

1. Upgrade the server by installing the Version 6.0 server software.
2. Upgrade your clients and applications over a period of time. As a temporary measure, you can continue to use clients running API Versions 3.0, 4.0 or 5.0 with the upgraded server..

If you have installed the Version 6.0 software on a client system, you cannot continue to run call processing applications compiled with a previous version of the CTC API. You must recompile your existing applications. If you do not do this, attempting to run the application will result in an Unsupported API Version error. For more information about this error, refer to the programming guides listed in Section 1.2.2.

When you upgrade your applications, you can make use of the new features available with API Version 6.0. For more information about these new features,

refer to the description of the API in the programming guides listed in Section 1.2.2.

Note that you cannot use clients running the Version 6.0 API software with a server running a previous version of the server software. You must upgrade the server software.

Part I

Installing and Configuring the Call Processing Server Software

Part I describes how to install, configure, and check the call processing server software:

- Chapter 2 tells you how to install the call processing server software.
- Chapter 3 describes how to configure the call processing server software and check the link to the switch.
- Chapter 4 tells you how to use the Authorization Setup Program to define authorized users.
- Chapter 5 describes the distributed data feature, and how to configure your call processing server(s) to use it.
- Chapter 6 describes how to use the Simple Object Access Protocol (SOAP) to pass Call Processing Software messages.
- Chapter 7 describes how to start and stop the call processing server.

Installing the Call Processing Server Software

2.1 Overview

This chapter describes how to install the call processing server software. The following table provides an overview of the contents of this chapter.

For details of...	See Section...
Kit contents	2.2
Hardware requirements	2.3
Software requirements	2.4
Instructions for attaching the hardware key	2.5
Installation instructions	2.6
Files installed	2.7
What to do after installation	2.8
Reinstalling the software	2.9
Removing the software	2.10

2.2 Kit Contents

The Call Processing Software V6.0 kit contains the following:

- CD-ROM
- A hardware license key
- Call Processing Software documentation set

The CD-ROM contains the call processing server software and installation kits for the call processing API software. An API kit is provided for each supported platform. You can choose to install all API kits or select the kits you require. The selected API kits are loaded into separate subdirectories on the call processing server.

The hardware key licenses your system to use the Call Processing Software. For more information, see Section 2.5.

2.3 Hardware Requirements

To install and run the call processing server software, you require:

- A PC suitable for installing Windows NT, Windows 2000 or Windows XP.
- At least 65 Mbytes of free disk space, without SOAP Access Components
or
At least 91 Mbytes of free disk space, with SOAP Access Components
For more information about SOAP Access Components, refer to Chapter 6.
- A CD-ROM drive.
- A network adapter card for connection to call processing clients. For suitable cards, refer to the documentation provided with your operating system, see your hardware supplier or refer to the Microsoft Web site (for example, to the *Windows Hardware Compatibility List*).
- A network adapter card for connection to the switch. The card that you install is dependent on the type of switch you are using and the network protocol used between the call processing server and the switch. Refer to Table 2–1 for details of supported cards.

Table 2–1 Supported Network Cards

Switch	Network Protocol	Network Card Required
CSTA switch		
	ISDN	DIVA card from Eicon† Technology. The Call Processing Software can be used with any of the cards in the DIVA range for Basic Rate Interface (BRI) ISDN.
	TCP/IP	A compatible network adapter card. Refer to the <i>Windows Hardware Compatibility List</i> for details.
	V.24	Network adapter card not required.
	X.25	An X.25 network adapter card available from Eicon Technology. X.25 is only available for links to a CSTA Phase I switch.
Avaya DEFINITY switches		
	ISDN	For details of hardware support, contact Intel.
	TCP/IP	A compatible network adapter card. Refer to the <i>Windows Hardware Compatibility List</i> for details.

Table 2–1 Supported Network Cards (Continued)

Switch	Network Protocol	Network Card Required
Nortel Meridian 1		
	TCP/IP	A compatible network adapter card. Refer to the <i>Windows Hardware Compatibility List</i> for details.
	X.25	An X.25 network adapter card available from Eicon Technology. X.25 is only available for links with AM Base Meridian Link.

2.4 Software Requirements

To install and run the call processing server software, you require:

- Windows NT (Windows NT Workstation or Windows NT Server) Version 4.0 (SP6), Windows 2000 (any version) or Windows XP (any version).
- Software for the network adapter card installed for communication with the call processing clients. For details, refer to the documentation provided with the network adapter card.
- Software for the network adapter card installed for communication with the switch. Refer to Table 2–2 for details of the software required.

Table 2–2 Supported Network Card Software

Card	Software Required
DIVA ISDN adapter card from Eicon Technology	<p>DIVA software for Windows NT, for example, from Eicon's ISDN Software Suite. Contact Eicon Technology for details.</p> <p>For Windows 2000/XP, you should download the required software from the Eicon Technology Web site: http://www.eicon.com</p> <p>Use the settings in Section 2.4.1 to configure the software for use with the Call Processing Software.</p>
TCP/IP network adapter card	The card manufacturer's Windows NT and Windows 2000/XP software drivers and application software. For details, refer to the documentation for the card.

Table 2–2 Supported Network Card Software (Continued)

Card	Software Required
X.25 network adapter card from Eicon Technology	EiconCard Connections for Windows NT and EiconCard Connections for Windows 2000/XP software. There are a number of components that you can install as part of the Eicon software. The Call Processing Software requires Core Services only. Use the settings in Section 2.4.2 to configure your EiconCard network adapter card for use with the Call Processing Software.

2.4.1 Configuring the DIVA ISDN Driver Software

To configure the DIVA ISDN driver software for use with the Call Processing Software, follow these steps:

1. Refer to your Eicon Technology documentation to start the ISDN driver software installation and configuration procedure and use the settings in Table 2–3 to configure the software.

Table 2–3 Configuring a DIVA ISDN Adapter Card

For this parameter...	Enter...
I/O Port Base Address	An appropriate setting for your Windows NT, Windows 2000 or Windows XP PC
Interrupt Number	An appropriate setting for your Windows NT, Windows 2000 or Windows XP PC
ISDN Protocol	For a Hicom [†] 150 E Office or Deutsche Telecom Octopus E 300/800 switch: Europe and other countries, EuroISDN (ETSI) For a Philips [†] SOPHO switch: Germany, national ISDN protocol [1 TR6]
Card Name	The name for your card

For all other configuration parameters, you can use the default settings.

2. When the ISDN driver software installation and configuration is complete, you are asked if you want to install the WAN Miniport driver. The Call Processing Software does not require this driver.

3. Install the Common ISDN Application Programming Interface (CAPI) V2.0 (supplied by Eicon Technology with the driver software). Follow the procedures in the Eicon Technology documentation for the driver software.

2.4.2 Configuring an EiconCard X.25 Network Adapter Card

To configure an EiconCard X.25 network adapter card for use with the Call Processing Software, follow these steps:

1. Start the WAN Services Configuration Program. Refer to your Eicon Technology documentation for details.
2. Set the values shown in Table 2–4. For all other options and parameters, you can use the default values offered by the configuration program.

Table 2–4 WAN Services Configuration Values

Option	Parameter	Setting
EiconCard hardware (for example, EiconCard C20)		
	Auto activate	Enabled (box checked).
High-Level Protocols		
	Transport ISO	Not required (disabled).
	Transport Bull TGX	Not required (disabled).
	Connection Manager	Not required (disabled).
Port (for example, Port1)		
	Name	A name for the port. Refer to the EiconCard documentation for the characters you can use in the name (for example, do not use spaces or underscores).
	Number	The number for the port
	Dialer	The appropriate setting for the connection between the call processing server and the switch, for example, Direct.
	Protocol	X.25

Table 2–4 WAN Services Configuration Values (Continued)

Option	Parameter	Setting
X.25		
	Node type	DTE
	Packet format	Basic
	X.25 Version	1984
	Window Size	
	Maximum	7
	Default	Intel recommends 7 for maximum transfer rate. However, this value must match the X.25 Window Size parameter on the switch. Check the value defined on the switch and, if possible, set both to 7.
	Packet Size	
	Maximum	256
	Default	256
	Sequential assignment of virtual circuits	Enabled.
	PVC Quantity	
	Meridian 1	0
	CSTA switch	1
	IVC Quantity	
	Meridian 1	0
	CSTA switch	0
	TVC Quantity	
	Meridian 1	1
	CSTA switch	0
	OVC Quantity	
	Meridian 1	0
	CSTA switch	0
	Data Link Layer	HDLC
HDLC		
General	FRMR to RRC/RNRC /REJC with P=0	Enabled

Table 2–4 WAN Services Configuration Values (Continued)

Option	Parameter	Setting
PVC (CSTA switches only)		
	No.	1
	Win size	The value specified for the Default X.25 Window Size (7 recommended).
	Packet size	256
Direct, Sync (if Direct selected as Dialer setting)		
General	Clocking	EXTERNAL

3. Save your parameter settings and exit the program.

2.5 Attaching the Hardware License Key

The Call Processing Software uses a hardware key to:

- Ensure that your system is licensed for running the server software.
- Enable the Call Processing Software functions your license provides.

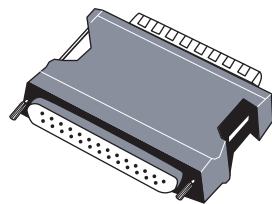
2.5.1 Types of Hardware Key

You can order one of the following types of hardware key with your Call Processing Software kit:

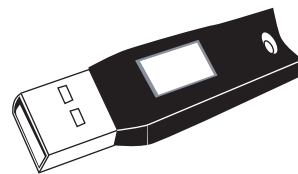
- A hardware key for a parallel printer port
- A hardware key for a USB port

Figure 2–1 shows both types of hardware key.

Figure 2–1 Call Processing Software Hardware Key



Parallel Port license key



USB license key

2.5.2 Attaching the Hardware Key

You need to do one of the following, depending on the type of hardware key in your kit:

- Attach the hardware key to the parallel printer port on your system.
- Attach the hardware key to the USB port on your system.

The hardware key must be present before you start the server software. Intel recommends that you attach it before you actually install the server software.

The hardware key should remain attached to the port. If the key is missing, the system handles the situation depending on the circumstances:

- If you attempt to restart a link between the server and a switch, the Call Processing Software logs event messages to your system to warn that the key is not attached and the link will not start.
- If you reboot the server or try to start the server software manually, the server software will only start in Evaluation Mode. In Evaluation Mode, the server software stops after two hours.
- If the periodic check of the key finds it is missing, or that the license key type has changed, the Call Processing Software logs this in the event log and prevents new links being started.

2.6 Installation Procedure

Follow these steps:

1. Make sure that your system has the required hardware and software configured and available on your system. See Sections 2.3 and 2.4.
2. Log in as administrator. You require administrator privileges to install the call processing server software.
3. Insert the Call Processing Software V6.0 CD-ROM into the CD-ROM drive.
4. Select **Run...** from the Start menu.
5. Enter `D:\SETUP` in the **Open:** text box, where *D* is the CD-ROM drive letter.
6. Click on the **OK** button.

The procedure:

- Prompts you to select the type of installation you require. You can select one of the following options:

Option	Description
Typical	Installs the call processing server software and copies all call processing API kits, online versions of the Call Processing Software documentation, and Acrobat Reader software to your system.
Compact	Installs the call processing server software.
Custom	Prompts you to select the software, call processing API kits, and documentation you require.

- Prompts you to select the type of interface(s) you require. You **must** select one or both of these options:

Option	Description
DCE /RPC	Enables the Distributed Computing Environment/Remote Procedure Calls (DCE/RPC) interface on your server. Enable this interface to support DCE/RPC-based call processing clients in your network. See Section 1.1 for a list of the DCE/RPC-based clients.
Java RMI	Enables the Java RMI interface on your call processing server. Enable this interface to support Java RMI-based call processing clients in your network. Java RMI-based clients are those running the Java Runtime Environment.

You can change your selection after the installation, using the Server Options in the Configuration Program.

- If you select a Typical installation, asks if you wish to install the SOAP Access Components on the server. You need these components only if you wish to provide external access to your call processing network through your firewall, as described in Chapter 6. Note that the SOAP Access Components include a Web server implementation (Jakarta Tomcat), and that running a Web server on your call processing server could compromise server security.
- Depending on the type of installation selected, installs the server software and loads copies of the API installation kits onto the server. Section 2.7 lists the location of the files installed on the server.

- If you are upgrading from an earlier version, asks you if you wish to migrate from the previous version. The V6.0 Call Processing Software database has changed its location from previous versions; if you answer yes to this question, your Call Processing Software registry entries will be moved to the new location.
- Asks if you want to read the release notes and/or run the Configuration Program.

When you run the Configuration Program, you must specify information about your switch and the connection to the server. Intel recommends that you gather this information before starting the program. Refer to Section 3.3 for details.

2.7 Files Installed

Table 2–5 lists and describes the Call Processing Software files installed on the call processing server after the installation is complete. In all cases, *drive:\directory* is the drive and directory chosen for the installation. By default, this is C:\Program Files\Intel\Ctc.

The number of files installed depends on the type of installation you chose. Note that the installation procedure always installs the Java Runtime Environment files on the call processing server.

Table 2–5 Files Installed on the Call Processing Server

Description	Location
Release notes	<i>drive:\directory</i>
Call processing server system and system support files	<i>drive:\directory\BIN</i>
Call processing API installation kit for HP-UX	<i>drive:\directory\CLIENTS\HPUX</i>
Call processing API installation kit for Java	<i>drive:\directory\CLIENTS\JAVA</i>
Call processing API installation kit for Solaris	<i>drive:\directory\CLIENTS\SOLARIS</i>
Call processing API installation kit for Tru64 UNIX	<i>drive:\directory\CLIENTS\TRU64</i>
Call processing API installation kit for OpenVMS	<i>drive:\directory\CLIENTS\VMS</i>
Call processing API installation kit for Windows	<i>drive:\directory\CLIENTS\W32</i>
Configuration files	<i>drive:\directory\CONFIG</i>
Adobe Acrobat Reader self-extracting EXE file	<i>drive:\directory\DOCS</i>
Copies of documentation in PDF format	<i>drive:\directory\DOCS\PDF</i>

Table 2–5 Files Installed on the Call Processing Server (Continued)

Description	Location
Support files for Java	<i>drive:\directory\JAVA</i>
Java Runtime Environment files	<i>drive:\directory\JRE</i>
Specific server-side Java support files	<i>drive:\CTCJAVA</i>
Jakarta Tomcat files	<i>drive:\CTCJAVA</i>

2.8 What to Do Next

When the installation is complete:

1. If you have not read the release notes, please read them now. The release notes may contain important information you should read before installing the call processing API. See Section 2.8.1 for details of how to access the release notes.
2. If you wish, refer to the information contained in the remainder of this chapter:
 - How to obtain and read the Call Processing Software online documentation (see Section 2.8.2 and Section 2.8.3).
 - How to reinstall the server software (see Section 2.9).
 - How to remove the server software from your system (see Section 2.10).
3. If you did not configure a link to the switch during the installation, gather the necessary information and run the Configuration Program now. Refer to Chapter 3.
4. If you configured a link to the switch during the installation, check that it is fully operational. Refer to Section 3.10.
5. Install the call processing API on your client system(s). See Part II of this guide.

2.8.1 Reading the Release Notes

When the installation is complete, SETUP asks if you want to display the release notes. If you answer Yes, the release notes are displayed in a Notepad window for you to read or print.

If you want to read the release notes at a later stage, you can display or print the file README.TXT. This file is copied to the drive and directory chosen for the installation. For example:

C:\Program Files\Intel\Ctc\readme.txt

You can also open the release notes by selecting the following from the Start menu:

Programs → Intel NetMerge Call Processing Server → Release Notes

Note that the Programs item on the Start Menu is All Programs on Window XP systems.

2.8.2 Reading the Online Documentation

If you copied the online documentation to your system as part of the installation, you can view, search, or print it using Adobe's Acrobat Reader. The Acrobat Reader 4.0 software is also copied to your system with the online files and, if Acrobat Reader is not already installed, you are asked whether you would like to install it. You can install it during the installation procedure by answering yes to the prompt, or you can install it later.

To install Acrobat Reader later, run the Acrobat Reader installation file:

drive:\directory\Docs\ACRD4ENU.EXE

To view the Call Processing Software documentation:

1. Start Acrobat Reader. From the Start menu open the following:

Programs → Adobe Acrobat → Acrobat Reader

2. Use Acrobat Reader to open the PDF file containing the documentation you wish to see. The PDF files are in the directory:

drive:\directory\docs\pdf

where *drive:\directory* is the drive and directory you specified for the call processing server software installation. For example:

C:\Program Files\Intel\Ctc\docs\pdf

You can also find the PDF files by selecting the following from the Start menu:

Programs → Intel NetMerge Call Processing Server → Documentation (PDF)

2.8.3 Reading the Java API Programming Documentation

The Java programming documentation for both the Java call processing API and the Java management API is in the files CTCAPIDOC.TARZ or CTCAPIDOC.ZIP (depending on your operating system). This documentation is in the form of HTML files.

You read the Java programming documentation with your web browser. To access the information, open the file INDEX.HTML from within your browser or double-click the INDEX.HTML file.

2.9 Reinstalling the Call Processing Server Software

If you need to reinstall the call processing server V6.0 software, you must first stop the call processing server service, CtcServer. Refer to Section 7.2 for complete instructions.

If you do not exit and stop the CtcServer service and you try to reinstall the product, the installation fails. For more information about starting and stopping the call processing server, refer to Chapter 7.

2.10 Removing the Call Processing Server Software

This section describes how to remove the call processing server software. Follow the steps appropriate for your operating system.

If you want to reinstall the software, reboot the server before starting the installation procedure.

2.10.1 Windows 9x and Windows NT

1. From the Control Panel, open the Add/Remove Programs dialog box.
2. Select the **Install/Uninstall** tab.
3. Select **Intel NetMerge Call Processing API** from the list of software.
4. Click on the **Add/Remove** button.

2.10.2 Windows 2000

1. From the Control Panel, open the Add/Remove Programs dialog box.
2. Click the Change or Remove Programs icon.
3. Select **Intel NetMerge Call Processing API** from the list of software.
4. Click on the **Remove** button.

2.10.3 Windows XP

1. From the Control Panel, Select Add or Remove Programs.
2. Select **Intel NetMerge Call Processing API** from the list of software.
3. Click on the **Remove** button.

Configuring the Call Processing Server Software

This chapter describes how to use the Call Processing Software Configuration Program to create, modify, or delete communications links between the call processing server and one or more switches.

3.1 Introduction

This section introduces the Configuration Program, and briefly describes the concepts you need to understand before you start configuring the software.

3.1.1 What is a Server Communications Link?

A server communications link is the logical communications path between the call processing server and a switch. The link identifies the type of network protocol used for communication, the network location of the two systems, the type of switching protocol used, and any additional information required for the two systems to exchange data.

You use the Configuration Program to configure a link and assign a name to the configured link. The link's name is known as the **logical identifier**. You use the logical identifier when you need to reference the link, or when the software requires you to identify the communications path between the server and the switch.

Once you have configured the link, it must be started. Starting a link prompts the server and the switch to negotiate communication so that the path is ready for exchanging data. By default, the link starts automatically when the server is started.

You can also start the link manually using the Call Processing Software Control Program; refer to Section 7.3.2. For more information about the Control Program, refer to the online help for the program.

3.1.2 The Configuration Program

The Configuration Program is a GUI application that you use to:

- Configure communications links between the server and one or more switches
- Modify the server options that affect the behavior of the call processing server (that is, are not limited to a specific link)
- Modify communications links
- Delete communications links

When you use the program, the link information you specify is stored in the Windows registry. When you delete a link, information about the link is deleted from the Windows registry. The server options you specify are also stored in the registry.

3.1.2.1 Authorization Setup Program

You can, if you wish, restrict access from call processing clients to telephony devices. To do this, you need to do both of the following:

- Enable Device Level Authorization in the Configuration Program.
- Configure authorized users in the Authorization Setup Program.

See Chapter 4 for details of how to use the Authorization Setup Program.

3.1.2.2 Distributed Data Setup Program

If your license supports distributed application data, there is an additional Distributed Data Setup Program for performing the required configuration. See Chapter 5 for details of how to use this program.

3.2 Configuring the Server and Creating Links: Procedure

To configure the server and links to switches, follow these steps:

1. Start the Configuration Program. From the Start menu on the server, select:
Programs → Intel NetMerge Call Processing Server → Configuration Program
2. Enter the information in the configuration windows. First enter a name for the link, under New Link on the first window. When you have entered a name, click the **Add** button. You will then go to a new window to select a switch type.

Refer to Tables 3–1 to 3–4 in Section 3.3 for more information.

3. Save the information you have entered and exit the Configuration Program.
4. If it is not already started, start the call processing server and start the link. Refer to Chapter 7 for information about starting a server and a link.
5. Check the link. Section 3.10 describes how to check a link.

3.3 Required Configuration Information

Before you start the Configuration Program, you need to gather configuration information. The information you need depends on your switch and the type of connection it uses to communicate with the call processing server (for example, TCP/IP, X.25, or ISDN).

Tables 3–1 to 3–4 contain details of the information required:

For this switch...	Refer to...
Switch supporting CSTA	Table 3–1 and Table 3–2
Avaya DEFINITY	Table 3–1 and Table 3–3
Nortel Meridian 1	Table 3–1 and Table 3–4

Note that Table 3–2 lists the generic configuration information required for all switches supporting CSTA. This covers a large number of switches, not all of which support all of the parameters listed. However, when you run the Configuration Program, it greys-out the parameters that are not supported by a switch, so you can see exactly which parameters are required.

Table 3–1 Configuration Information for All Switches

Information	Description and Default Values
Logical Identifier	<p>The name that you use to identify the link between the server and the switch. The logical identifier can only be made up of the characters A-Z, a-z, 0-9, and the underscore (_). For example, NCP1. Other characters, for example, asterisk (*) and hyphen (-) are not supported. The maximum length is 15 characters.</p> <p>Once you click the Add button, you go to the Switch Type and Transport window.</p>
Server Options button	<p>Click the Server Options button to configure options for Management Tracing, Server Interfaces and Remote Management. Server options are not link specific but apply to all links and clients associated with this server. The options are explained under Server Options on page 3-8.</p>
<hr/> Switch Type and Transport <hr/>	
Switch Type and Transport	<p>The name of the switch and the transport (network) protocol used for the connection between the server and the switch.</p> <p>The Configuration Program displays a list of supported switches and the protocols supported for each. Select your switch and protocol from the list.</p> <p>Click the Next button to go to the Configuring Link window.</p>
<hr/> Configuring Link <hr/>	
Maximum Monitors	<p>The number of logical channels that can be monitored by call processing applications at any one time.</p> <p>The default is 1000. However, the maximum number of monitors you should specify is dependent on:</p> <ul style="list-style-type: none">• The limitations defined by your Call Processing Software license. See your <i>Intel NetMerge Call Processing Software Customer Letter</i> for details.• The system resources available on your server.• Any limitations defined by your switch (for details, see your switch administrator). <p>If you specify more monitors than these limits allow, your application may not be able to use them or your system may run out of resources.</p>
Auto Start Link	<p>If you check this box, the link automatically starts up when the server is booted. Default: Enabled (the box is checked).</p> <p>Intel recommends that you enable Auto Start so that the specified link is started automatically. To start the link manually, use the Control Program; see Section 7.3.2.</p>

Table 3–1 Configuration Information for All Switches (Continued)

Information	Description and Default Values
Switch Timestamp	<p data-bbox="656 470 1360 548">Specifies whether the timestamp returned by the call processing API is the one generated by the switch or the one generated by the server.</p> <p data-bbox="656 569 1360 646">Default: The timestamp generated by the server (that is, the box is not checked). If you want the call processing API to return the timestamp from the switch, you must check the box.</p> <p data-bbox="656 667 1360 768">Note that some switches and protocol configurations do not support the switch timestamp. The Configuration Program greys out the check box if the switch timestamp is not supported by your configuration.</p> <p data-bbox="656 789 1360 884">The call processing API returns timestamp information with the <code>ctcGetEvent</code> routine or the <code>ctcEventListener</code> interface methods. Refer to the programming guides listed in Section 1.2.2 for more information.</p>
Auto Restart Monitors	<p data-bbox="656 905 1360 982">If you check this box, device channels are automatically re-established, and monitoring started, when a link is restarted after having previously been down.</p> <p data-bbox="656 1003 1360 1155">Note: If you want to use this feature, and you are using a Version 5.0 or earlier client application, you may need to modify your application. This is because enabling this feature may affect the way your application handles link restarts and establishing monitor channels. Refer to the programming guides listed in Section 1.2.2 for more information.</p>

Table 3–1 Configuration Information for All Switches (Continued)

Information	Description and Default Values
Device Level Authorization	<p>Device Level Authorization (DLA) lets you control user access to telephony devices. If you want to use this feature, you must:</p> <ul style="list-style-type: none">• Check the Device Level Authorization box.• Select the type of password (if any) that the call processing server should look for when checking user authorization in the authorization database:<ul style="list-style-type: none">– None. The server does not look for a password. It authorizes access on the basis of the login user name on the client system.– System. The server verifies the password (and user name) passed by the assign channel request against those set up for the user either on the domain server or in a local user account on the call processing server. It then checks the user name against the one set up in the authorization database.– Application. The server checks the password (and user name) passed by the assign channel request against those set up in the authorization database. Refer to Chapter 4 for details. <p>The System and Application options will only work if the client API has been upgraded to V6.0, and the client application passes a user name and password in its assign channel request.</p> <p>In addition to enabling DLA here, you must define a list of authorized users. See Chapter 4 for full instructions on defining authorized users.</p> <p>Note: If you enable DLA on a link, then the only users allowed access to devices via that link will be those defined as authorized users.</p>
<hr/> Trace <hr/>	
<p>If you click the Trace button on the Configuring Link window, you go to the Trace File Settings window. You can supply information for any or all of four trace levels:</p>	
<ul style="list-style-type: none">• CTC - Traces application (API) function results.• CTCFULL - Traces API requests and results.• NET - Traces protocol data on the link, including any protocol-specific transport data.• DIST - Provides server to server tracing in a distributed data environment.	
Auto Start Trace	<p>Check this box if you want tracing to start automatically when the link is started.</p>

Table 3–1 Configuration Information for All Switches (Continued)

Information	Description and Default Values
Maximum Files	Specifies the maximum number of trace files to be used for tracing at the specified level. Default: 999.
Maximum Files Action	Specifies what should happen when all trace files are full. Select Stop if you want the system to stop tracing. Select Overwrite if you want the system to start overwriting the data in the oldest trace files.
Trace File Path	The path for trace files at the specified trace level. The default directory is <i>drive\directory\traces</i> , where <i>drive\directory</i> is the installation directory. The default installation directory is C:\Program Files\Intel\ctc.
Trace File Name	The trace file name for the specified trace level. You can change the default if you wish. The default trace file name is in the format <i>logid_tracelevel.n</i> , where <i>logid</i> is the logical identifier for the link, <i>tracelevel</i> is the trace level and <i>n</i> is the sequence number. For example, <i>parislink1_ctcfull.001</i> .

Advanced

Click the **Advanced** button on the Configuring Link window to go to the Advanced Settings window. **Note:** You are advised not to adjust these settings unless you fully understand their potential impact on call processing links and on system resources. The Advanced settings are described in more detail in Section 3.4.

Link State Checking

Enable Link State Checking	Check this box to enable link state checking on the link. Default: Disabled.
Check Interval	Specifies the frequency with which the link state is checked. Range: 10 and 300 seconds. Default: 30 seconds.
Retry Count	Specifies how many times the call processing server retries if the switch does not respond to a link state check. Range: 1 to 5. Default: 2.

Event Buffering

Device Buffers	Specifies the number of buffers allocated to hold DN device events. Range: 6 to 100. Default: 6.
Monitor Channel Buffers	Specifies the number of buffers allocated to hold monitor channel events. Range: 20 to 100. Default: 20.
Route Point Buffers	Specifies the number of buffers allocated to hold route point requests. Range: 6 to 100. Default: 6.

Data Parameters

Swap Call References	Check this box to select the call reference format you require. The default is network byte order.
----------------------	--

Table 3–1 Configuration Information for All Switches (Continued)

Information	Description and Default Values
Server Application Data	Check this box to specify that the server should store application data associated with a call. Default: Disabled.
Server Options	
Click on the Server Options button on the first configuration window to go to the Server Options window.	
Management Tracing	
Auto Start Trace	Automatically starts management tracing when the server reboots. Management tracing covers Control Program and management API messages.
Maximum Files	Specifies the maximum number of trace files to be used to hold management traces. Default: 999.
Maximum Files Action	Specifies what should happen when all management trace files are full. Select Stop if you want the system to stop tracing. Select Overwrite if you want the system to start overwriting the data in the trace files.
Trace File Path	The path for management trace files. The default directory is <i>drive\directory\traces</i> , where <i>drive\directory</i> is the installation directory.
Trace File Name	The management trace file name. The default is <i>mgmttrace.n</i> where <i>n</i> is the number of the trace file in the sequence.
Server Interfaces	Select the interfaces you need to support the clients in your network. Choose DCE/RPC, Java RMI or both. Default: The selection made during installation. Refer to Section 1.2.2 for an explanation of these interfaces.
Client Security for Remote Management	Specify any client systems and users that are allowed to remotely manage the server. By default, no client is permitted to do this. If you want remote management, you must explicitly configure the security parameters here. Click the Add button to add a new client and user.
Host Name	Enter the name of the client system from which you will allow a user to manage the call processing server.
User Name	Specifies the user name of the person you will allow to manage the call processing server.
Security Level	Gives the level of access you will allow the person specified in the User Name box. Monitor only allows the user to display information about the server. Full allows the user to modify the server operation (for example, using the Control Program or the Configuration Program).

Table 3–2 Configuration Information for CSTA Switches

Information	Description and Default Values
All Connections	
Device Query	<p data-bbox="662 506 1357 638">Specifies whether the server queries the switch about the validity of the device number when a channel is assigned. A call processing application must assign a channel to a device (for example, telephone) before it can control and monitor calls to and from that device.</p> <p data-bbox="662 657 1357 814">When a switch is known to the Configuration Program, it enables or disables Device Query by default. However, if you are configuring an unnamed switch (that is, using the <i>CSTA Phase I Other</i>, <i>CSTA Phase II Other</i> or <i>CSTA Phase III Other</i> switch types), check with your switch administrator whether Device Query is or is not available.</p> <p data-bbox="662 833 1357 966">If you check the Device Query box, the server queries the validity of specified device numbers (for example, telephone numbers, Directory Numbers (DNs) or route points). If you do not check the box, Device Query is disabled and the server does not query the validity of device numbers.</p> <p data-bbox="662 984 1357 1083">If your switch supports it, Intel recommends that you enable Device Query to ensure that your call processing application can determine whether a user has entered an invalid value for their extension or telephone number.</p>
ACSE	<p data-bbox="662 1102 1357 1150">Specifies whether the switch uses the Association Control Service Element (ACSE) protocol.</p> <p data-bbox="662 1169 1357 1325">When a switch is known to the Configuration Program, it enables or disables ACSE by default. However, if you are configuring an unnamed switch (that is, using the <i>CSTA Phase I Other</i>, <i>CSTA Phase II Other</i> or <i>CSTA Phase III Other</i> switch types), check with your switch administrator whether ACSE should or should not be enabled.</p>

Table 3–2 Configuration Information for CSTA Switches (Continued)

Information	Description and Default Values
Call Reference Length and Call Reference Type (Philips SOPHO switch only)	<p>The meaning of the call reference length depends on the format you indicate in call reference type.</p> <p>Set the call reference type to string if the call reference length is greater than 4. Set the call reference type to integer or string if the call reference length is 4 or less. The default is string.</p> <ul style="list-style-type: none">• For integer, the call reference length you enter specifies the size of the integer. The default is 4.• For string, the call reference length you enter specifies the number of characters in the string (not counting the terminating NUL). The default is 4. <p>Typically, you can accept both defaults (4 and string) unless your switch is in a network of switches. In this case, see your switch administrator for details of the switch's call reference format.</p>
TCP/IP Connections	
Switch IP Address	<p>The IP address of the switch. Enter this using:</p> <ul style="list-style-type: none">• The format <i>n.n.n.n</i>, where <i>n</i> is a number between 0 and 255. For example, 24.36.0.189.• A name already defined in a Host File or in a Distributed Name Server (DNS) database <p>Ask your switch administrator for the address that you specify.</p>
Port Number	<p>The TCP/IP port number that the call processing server uses to communicate with the switch.</p> <p>You must set the TCP/IP port number here to be the same as the TCP/IP port number on the switch. The link between the server and the switch will only work if the TCP/IP port numbers are identical.</p>
Local IP Address	<p>The IP address for the call processing server network adapter card. Specify an address only if there is more than one adapter card installed and you need to identify the card used for the connection to the switch.</p>
Secondary IP Address	<p>A secondary IP address for the switch. Specify an address only if your switch supports secondary IP addressing. The call processing server can use the secondary IP address to establish a link to the switch if the link to the primary IP address fails.</p>

Table 3–2 Configuration Information for CSTA Switches (Continued)

Information	Description and Default Values
ISDN Connections	
Device	The name of the port on the ISDN network adapter card used for the connection to the switch. If you have only one link to the switch, accept the default name ECP1. If you are configuring a second link, enter the name ECP2.
Dialup Number	A number configured on the switch used to establish the connection from the call processing server. Specify the number defined on the switch. You need to obtain this number from your switch administrator.
X.25 Connections (CSTA Phase I switches only)	
Device	The name of the port on the X.25 network adapter card being used for the PVC link to the switch. If you have only one link to the switch, accept the default name ECP. If you are configuring a second link enter the name ECP1.
LCN	The Logical Channel Number for the X.25 PVC that is used for the link to the switch. See your switch administrator for the value you specify. The default value is 1.
V.24 Connections	
Device	The name of the port used to attach the cable for connection to the switch. The default is COM1.
Baud Rate	The speed of the connection. The default is 9600 baud.

Table 3–3 Configuration Information for Avaya DEFINITY Switches

Information	Description and Default Values
TCP/IP Connections	
Switch IP Address	<p>The IP address of the switch or the LAN Gateway/MAPD module. Enter this using:</p> <ul style="list-style-type: none">• The format <i>n.n.n.n</i>, where <i>n</i> is a number between 0 and 255. For example, 24.36.0.189.• A name defined in a Host File or in a Distributed Name Server (DNS) database. <p>Ask your switch administrator for the address that you specify.</p>
Link Number	<p>A number from 1 through 8 that is set by the switch administrator and used to uniquely identify the communications link to the Avaya DEFINITY. See your switch administrator for the number that you specify.</p> <p>If you configure more than one link from the call processing server to the same Avaya DEFINITY, each link must have a unique link number. The default is 1.</p>
Local IP Address	<p>The IP address for the call processing server network adapter card. Specify an address only if there is more than one adapter card installed and you need to identify the card used for the connection to the switch.</p>
ISDN Connections	
Device	<p>The name of the port on the ISDN network adapter card used for the connection to the switch. Accept the default value APC1.</p>

Table 3–4 Configuration Information for Meridian 1 Switches

Information	Description and Default Values
TCP/IP Connections	
Note: You can configure multiple links over TCP/IP connections between the call processing server and a Meridian 1 switch. You must specify a unique application identifier for each link.	
Switch IP Address	The C-LAN address of a Symposium system or the IP address of the Meridian Link Module. Enter this using: <ul style="list-style-type: none">• The format <i>n.n.n.n</i>, where <i>n</i> is a number between 0 and 255. For example, 24.36.0.189.• A name defined in a Host File or in a DNS database. Ask your switch administrator for the address that you specify.
Port Number	The TCP/IP port number that the call processing server uses to communicate with the switch. Do not change the default value of 3000.
Switch Option	Enter 1 to ensure that the <code>ctcK_OpAnswered</code> call event will contain information identifying the other party. The equivalent event for Java programs is <code>EV_OP_ANSWERED</code> . Refer to the programming guides listed in Section 1.2.2 for more information about this event.
Local IP Address	The IP address for the call processing server network adapter card. Specify an address only if there is more than one adapter card installed and you need to identify the card used for the connection to the Meridian 1.
Application Identifier	Mandatory. You must specify this name to uniquely identify the link to the Meridian 1 switch. The application identifier can be any name of up to 20 alphanumeric characters. For example, <code>merilink1</code> . Each link to the same Meridian 1 switch must have a unique application identifier.
Machine Identifier	A name that identifies the type of Meridian 1 switch that you are using. Do not change the default identifier.
Customer Number	An identification number for your customer group on the switch. If your switch is partitioned to accommodate different groups of users (or customers), ask the switch administrator to supply the appropriate customer number. The program provides 0 as the default.
Host Name	The name that identifies the computer host to the switch, assigned during the switch configuration. Do not change the default name.

Table 3–4 Configuration Information for Meridian 1 Switches (Continued)

Information	Description and Default Values
Meridian Mail Name	<p>The name of the Meridian Mail system used to provide voice processing. This value is required only if a call processing application supports Meridian Mail features.</p> <p>Note that if you enable the Meridian Mail option and your switch setup does not include a Meridian Mail system, it may cause communication problems between the call processing server and your switch. Make sure that you do not enable the Meridian Mail option unless it is required.</p> <p>Ask the switch administrator for the name of the Meridian Mail system. MeridianMail is provided as the default.</p>
X.25 Connections	
Remote DTE Address	The remote DTE address is the DTE address for the Meridian 1 switch. Ask the switch administrator for the DTE address that you specify. The default address is 000000990100.
Local DTE Address	The local DTE address is the DTE address for the call processing server. Do not change the default value of 1111.
Switch Option	Enter 1 to ensure that the ctcK_OpAnswered call event will contain information identifying the other party. The equivalent event for Java programs is EV_OP_ANSWERED. Refer to the programming guides listed in Section 1.2.2 for more information about this event.
Application Identifier	The application identifier is a name that you specify to uniquely identify the link to the Meridian 1 switch. The application identifier can be any name of up to 20 alphanumeric characters.
Machine Identifier	The machine identifier is a name that identifies the type of Meridian 1 switch that you are using. Do not change the default identifier.
Customer Number	The identification number for your customer group on the switch. If the switch is partitioned to accommodate different groups of users (or customers), ask the switch administrator to supply the appropriate customer number. The default customer number is 0.
Host Name	The host name identifies the computer host to the switch and is assigned during the switch configuration. Do not change the default name.
Device	The name of the port on the X.25 network adapter card being used for the SVC link to the switch. If you have only one link to the switch, accept the default name ECP1. If you are configuring a second link enter the name ECP2.

Table 3–4 Configuration Information for Meridian 1 Switches (Continued)

Information	Description and Default Values
Meridian Mail Name	<p>The name of the Meridian Mail system used to provide voice processing. This value is required only if a call processing application supports Meridian Mail features.</p> <p>Note that if you enable the Meridian Mail option and your switch setup does not include a Meridian Mail system, it may cause communication problems between the server and your switch. Make sure that you do not enable the Meridian Mail option unless it is required.</p> <p>Ask the switch administrator for the name of the Meridian Mail system. MeridianMail is provided as the default.</p>

3.4 Advanced Settings

This section gives a detailed description of the parameters on the Advanced Settings window. To go to this window, click the **Advanced** button on the Configure Link window.

Normally, you do not need to adjust these settings. They are provided to offer you more control of a link for specific purposes, as described below. Please do not adjust these settings unless you fully understand their potential impact on call processing links and on system resources.

3.4.1 Link State Checking

If you enable link state checking, the server periodically sends a message to the switch to check that the link is up. If there is no reply after 30 seconds, the server will retry before defining the link as down. (**Note:** the server will then shut down the link and initiate the link restart procedure.)

You can define how many times the server retries if the switch does not respond to a link state check. You might do this if you wished to check the link more often than is provided for by the link protocol. For example, the TCP/IP protocol by default only checks the link every two hours.

Caution: Link state checking will only work if your switch has the ability to respond to a system status message. For example, if you have a CSTA switch, it must support the CSTA System State Request and Response messages. If you enable link state checking, and your switch does not support the required messages, you will log failures every time the link is checked, and the link will restart.

3.4.2 Event Buffering

The call processing server allocates buffers to hold recent device and monitor channel events. Your call processing application retrieves these using the `ctcGetEvent` routine or the `addEventListener` method.

The event buffering setting allows you to increase the number of buffers allocated. If there are not enough buffers, your application may lose events. This might happen if large numbers of events were coming over the link at the same time.

Refer to the programming guides listed in Section 1.2.2 for more information about processing events.

3.4.3 Route Point Buffering

The call processing server allocates buffers to hold route point requests. Your call processing application retrieves these using the `ctcGetRouteQuery` routine or the `addRouteListener` method.

The route point buffering setting allows you to increase the number of buffers allocated. You might want to do this if you expect a large number of route point requests to come over the link at the same time.

Refer to the programming guides for more information about processing route point requests.

3.4.4 Data Parameters

3.4.4.1 Swap Call References

If your switch is a Nortel Meridian 1, you can enable swap call references to change the order in which the Call Processing Software presents a call reference identifier to your application. By default, the software presents Nortel Meridian 1 call reference identifiers in network byte order. However, if your application requires it, the *Swap Call References* option causes the software to present call reference identifiers in Meridian Link Protocol Call ID format.

3.4.4.2 Server Application Data

Use this option to enable the server to store application data associated with a call. You might want to do this if application data is not supported by your switch. Note that if you enable server application data and your switch also supports it, data is stored on the server but data from the switch will take precedence.

If you wish to use the distributed data feature, you should enable this option for each link that will handle distributed data. You must also run the Distributed Data Setup Program to configure local and remote links for distributed data; refer to Chapter 5. Note that not all licenses support distributed data.

3.5 Maximum Monitors

Use the Control Program to temporarily change the maximum number of monitor channels available over a link. The maximum number of monitors is the only configuration setting that can be changed using the Control Program.

Follow these steps:

1. Start the Control Program. From the Start menu on the server, select:
Programs → Intel NetMerge Call Processing Server → Control Program.
2. Click the **Monitors** button on the main window.
3. Change the number for maximum monitors, and click the **OK** button.

3.6 Configuring Additional Links

You can use the Configuration Program to configure links between a call processing server and different switches or, if your switch supports multiple links, several links from the server to the same switch.

If you configure additional links, note that:

- Each link must have a unique logical identifier.
- If you configure more than one link to the same switch, certain values set for the switch must be unique. For details of these values, refer to Tables 3–1 to 3–4.

For example, if you configure two links between the server and a Meridian switch, each link must have a different application identifier.

3.7 Modifying a Link

If you need to change the configuration of a link, follow these steps:

1. If the link is running, use the Control Program to stop the link. To start the Control Program, from the Start menu on the server, select:
Programs → Intel NetMerge Call Processing Server → Control Program.
2. Select the link you want to stop from the list of logical identifiers.

3. Click the **Off** button.
4. Start the Configuration Program. From the Start menu on the server, select:
Programs → Intel NetMerge Call Processing Server → Configuration Program.
5. From the pull-down menu, select the logical identifier for the link you want to modify and click the **Modify** button.
You will see the Configure Link window. You can now change details of the link.
6. If you wish, click the **Advanced** button to change Advanced settings or the **Trace** button to change trace settings. Note that you must click on the **Save** button in these windows to enable your settings.
7. Save your changes and exit the Configuration Program.
8. Now, start the link from the Control Program. If the Control Program is not started, start it, as described in step 1.
9. On the screen, select the link you want to start from the list of logical identifiers.
10. Click the **On** button.

3.8 Deleting a Link

To delete a link:

1. Start the Configuration Program, as described in Section 3.2, Step 1.
2. From the pull-down menu, select the logical identifier for the link you want to delete and click the **Delete** button.

When you click the **Delete** button, all details of the link are deleted from the Windows registry.

3.9 Leaving the Configuration Program

To leave the Configuration Program, click on the **Exit** button.

3.10 Checking the Communications Link to the Switch

To check the link between the call processing server and a switch, use the Control Program, as follows:

1. If the call processing server is not already running, start the server and the link. Refer to Section 7.3 for information on starting the server and the link.
2. Start the Control Program. From the Start menu on the server, select:
Programs → Intel NetMerge Call Processing Server → Control Program.
3. Check the list of displayed links. The display shows the state of each link.
 - If the link's logical identifier is listed, the switch is known to the server. The state of the link should be ON. If the link is in a different state, use the diagnostic procedures in the online Problem Solving and Monitoring Help to solve the problem. From the Start menu on the server, select:
Programs → Intel NetMerge Call Processing Server → Problem Solving and Monitoring Help.
 - If the logical identifier is not listed, the link associated with that identifier is not configured. Return to Section 3.2 and follow the instructions to configure the link.

Refer to the Control Program online help for more information about showing links.

Configuring Device Level Authorization

This chapter describes how to configure user information for device level authorization (DLA).

4.1 Introduction

This section describes the device level authorization feature and summarizes the main configuration steps.

4.1.1 What is Device Level Authorization?

Device level authorization is an access control feature that enables you to control user access to telephony devices. In order to use DLA, you need to configure authorized users, and enable DLA checks on one or more links on the call processing server.

When you configure authorized users, the information is stored in an authorization database on the call processing server. The information includes:

- The user name to be checked on a given link
- The telephony devices to which each user is allowed access on the link
- The type of access which each user is allowed for each device
- The client system(s) from which each user is allowed access

4.1.2 Configuring Device Level Authorization

Configuring DLA is a two part process:

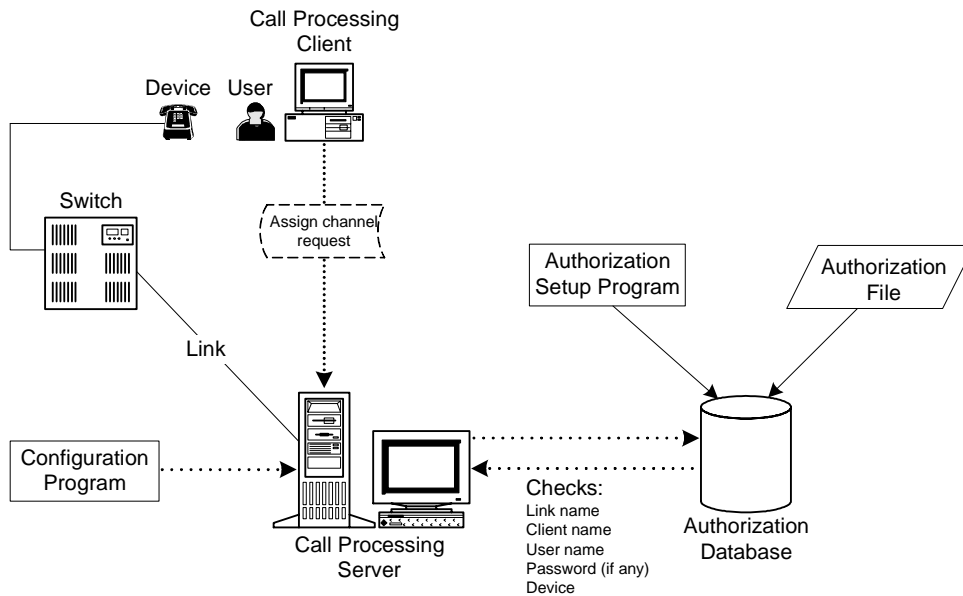
- Use the Configuration Program to enable DLA on a link, and specify the type of password checking the server will use; refer to Chapter 3, Table 3-1.
- Define authorized users in the authorization database. To do this, you use either or both of the following:
 - The Authorization Setup Program, described in Sections 4.2 to 4.4.
 - The Import/Export utility, described in Section 4.5.

Note: If you enable DLA on a link in the Configuration Program, then you *must* define authorized users, using either the Authorization Setup Program or the Import/Export utility. If you do not do this, then *no* users will be able to access devices associated with that link.

4.1.3 Checking User Authorization

Figure 4–1 shows how the call processing server checks user authorization, and where it gets the authorization information.

Figure 4–1 Checking User Authorization



In order for a user to access a device through a call processing application, the application requests that a channel be assigned to that device. When the server receives the channel request, it does the following:

1. Determines whether DLA is enabled on the link specified in the request, using information entered in the Configuration Program.
2. If DLA is enabled, checks the user information in the request against the user information in the authorization database.

Note: The term channel request refers here to the `ctcAssign` routine (procedural API) or to the constructor for creating a `CtcChannel` object (Java API).

4.1.4 Upgrading Client Applications for Device Level Authorization

If you are using a CTC Version 5.0 or earlier application with a Call Processing Software Version 6.0 server, and you want the server to check passwords, you need to:

- Upgrade the call processing API to Version 6.0.
- Upgrade your application to allow users to enter user names and passwords.

For more information about user names and passwords, refer to Sections 4.2.1 and 4.2.2. For more information about Version 6.0 functions, refer to the call processing API programming documentation.

4.2 Authorization Setup Program: Information Required

This section describes the information you supply in the Authorization Setup Program.

4.2.1 User Name

The type of user name you need to supply is determined by the Password option you selected in the Configuration Program, as described in Section 4.2.1.1. Refer to Chapter 3, page 3-6 for more information about selecting the password option.

The user name can have up to 64 alphanumeric characters.

4.2.1.1 User Names and Password Checking Options

When you enable Device Level Authorization on a link in the Configuration Program, you must select a password type. The type of user name you configure for authorization is determined by the password type you selected, as follows:

Password Type	User Name Type
None	The login user name on the client system. See Section 4.2.1.2.
System	A name that identifies a user in a Windows domain. You can use either of the following: <ul style="list-style-type: none">• A user name held on a domain server. See Section 4.2.1.3.• A user name in the local user accounts on the call processing server. See Section 4.2.1.4.
Application	A name that identifies a user specifically for the call processing application. See Section 4.2.1.5.

4.2.1.2 Login User Name (Password: None)

If you selected None for password checking, enter the user's login name on the client system. Do not include the domain name.

4.2.1.3 Domain User Name (Password: System)

This is a user name held on the domain server. If you want to use a domain user name, enter both the name of the domain and the user name, as follows:

domain\username

For example, Eurosales\srobinson

When the call processing server receives a domain user name and password in the assign channel request, it does the following:

- Finds the domain server on the network, and verifies the user name and password on the domain server.
- Checks the user name and password against those in the authorization database.

4.2.1.4 Local User Name (Password: System)

This is a user name held in the local user accounts on the call processing server. If you want to specify a local user name, just enter the user name, as follows:

username

For example, SROBINSON

To configure local user accounts on the call processing server, use the Control Panel (Local Users and Groups on Windows 2000 and Windows XP systems; Users on Windows NT systems). Refer to the Windows documentation for more information.

If you specified a System password, and you enter a user name without a domain, the call processing server assumes that it should verify the name in its local user accounts, and that the local user's domain is the call processing server's own local domain.

When the call processing server receives a local user name and password in the assign channel request, it does the following:

- Verifies the user name and password in its local list.
- Checks the user name and password against those set up in the authorization database.

4.2.1.5 Application User Name: (Password: Application)

The application user name only needs to match the name entered in the assign channel request; it does not need to be a Windows user name.

When you enter an application user name in the Authorization Setup Program, use this format:

username

For example, SUSANROBINSON

When the call processing server receives an application user name and password in the assign channel request, it checks them against the user name and password in the authorization database.

4.2.1.6 Client Applications and User Name and Password Checking

If you are using either System or Application password checking, you must include the user name and password in the assign channel request. Refer to the programming documentation for details.

4.2.2 Password

If you selected Application for the password type in the Configuration Program, you *must* configure a password for a user in the Authorization Setup Program.

If you selected either None or System for the password type in the Configuration Program, do not configure a password for a user in the Authorization Setup Program; it will be ignored.

The password can have up to 32 alphanumeric characters.

Note that all passwords are automatically encrypted.

4.2.3 Devices

For each user, supply information about the devices the user is permitted to access. The information is described in Sections 4.2.3.1 to 4.2.3.5.

You can add any number of devices for each user.

4.2.3.1 Device Type

You need to select a device type from the list in Table 4–1.

Table 4–1 Device Types

Device Type	Description
DN	A telephony device, such as a telephone, multiline set, group (queue) or Voice Response Unit (VRU).
Route point	A route point. A route point is a logical device used for call routing. It has a dialable number but there is no physical device.
Physical Id	A physical identifier. This identifies a telephony device by its switch reference, for example, a trunk identifier. This is only supported for CSTA switches.
Call	This allows you to monitor and/or control calls (based on call reference numbers), rather than the devices associated with the calls. This is only supported for DEFINITY switches.
Agent	An ACD agent. This is only supported for Meridian switches.

4.2.3.2 Device Number

The number you enter depends on the device type, as shown in Table 4–2.

Table 4–2 Device Numbers

Type	Device Number Format
DN	The directory number (telephone number) of the device. This is an ASCII string that can contain any combination of numbers 0 through 9 plus additional characters supported by your switch (usually * and #).
Route point	The telephone number of the route point. This is defined as for DN.
Physical Id	A switch reference, for example, a trunk identifier. Specify this number in hexadecimal digits.
Call	Do not enter a device number for this device type. If you select Call, it means that the user is authorized to control or monitor all calls, rather than a specific call.
Agent	The agent's position ID. The position ID uniquely identifies an agent in an ACD group (ACD queue), and is defined by the switch administrator.

Specifying Multiple Device Numbers

You can use a wild card within a device number to specify multiple device numbers. The only valid wild card is ? (question mark). For example:

200? specifies all numbers starting with 200

?001 specifies all numbers ending with 001

40?00 specifies all numbers starting with 40 and ending with 00

Note that you can only use one wild card within a device number.

If you want to enter a number range that cannot be specified using wild cards, then check the Range box, as described in Section 4.2.3.3.

4.2.3.3 Range

If you want to enter a range of device numbers, check the Range box on the Devices screen. This allows you to enter groups of numbers that cannot be covered using wild cards; for example, 185 to 201. When you check the Range box, you see two device number boxes, one for the starting number and one for the ending number.

4.2.3.4 Link Name

Select the logical identifier for the link associated with this device.

All configured links are listed in the list box, whether or not they have been enabled for DLA in the Configuration Program (see Table 3-1, page 3-6). If you select a link that has not been enabled for DLA, the server will not do any authorization checks until you have enabled the link for DLA.

4.2.3.5 Permission Level

The permission level indicates the type of access allowed for the user to this device. The permission levels are:

- Full. This allows all calls from the user to the device.
- Monitor. This only allows the user to monitor the device.

4.2.4 Client System Names

For each user, you need to specify the host name of each client system from which the user will be allowed to access devices.

4.3 Using the Authorization Setup Program

The general procedure for setting up authorized users is as follows:

1. Start the Authorization Setup Program. From the Start Menu, select:
Programs → Intel NetMerge Call Processing Server → Authorization Setup Program.
2. Click the **Add** button for the User List. In the pop-up window, enter a user name, and click the **OK** button.

3. (Optional). If you want to enter a password for the user, click the **Password** button. In the pop-up window, enter a password. Enter it again to confirm the password, and click the **OK** button.
4. With the user selected, click the **Add** button for Devices. In the pop-up window, enter the device information (listed in Section 4.2.3). Click the **Apply** button.
5. Repeat Step 4 for each device that you want the user to access. After you have clicked the **Apply** button for the last device, click the **Cancel** button. to return to the main window.

If you only want to enter one device, you can enter the device and then click the **OK** button to return to the main window.
6. Click the **Add** button for Systems. In the pop-up window, enter a client system name and click the **OK** button.
7. Repeat Step 6 for each client that you want the user to be allowed to use.
8. Repeat Steps 2 to 6 for each user that you want to define.
9. Click the **Exit** button to leave the program.

4.4 Authorization Setup Program: Deleting and Modifying Information

4.4.1 Deleting Information

To delete a user, device or client:

1. Select the user, device or client that you want to delete.
2. Click the **Remove** button to the right of the appropriate list.

4.4.2 Modifying Device Information

To modify device information, follow these steps:

1. Select a user.
2. Under **Devices**, select the link associated with the device.
3. Click on the plus sign to see the devices, and select a device to modify.
4. Click the Edit button. You will see the Device screen.
5. On the Device screen, make your changes.
6. Click the **Apply** button.

4.4.3 Modifying a Password

To change a user's password, follow these steps:

1. Select the user whose password you want to change.
2. Click on the **Password** button.
3. Enter the new password.

4.5 The Import/Export Feature

The import/export feature allows you to import or export an *authorization file* to or from the authorization database. An authorization file is a text file holding information about authorized users, in the form of keywords and data. The information represented in this file is the same as the information that you enter in the Authorization Setup Program.

You can use this feature in two ways:

- **Importing** an authorization file allows you to add a list of users to the authorization database without configuring each user separately in the Authorization Setup Program. You can, if you wish, create your own authorization file, say, from a company telephone directory.
- **Exporting** an authorization file from the existing authorization database allows you to back up authorization information. You can also edit the exported file, and then re-import it.

4.5.1 How to Import or Export

There are two ways to import or export an authorization file:

- Using the **Import** or **Export** button in the Authorization Setup Program.
- Using the Import/Export utility. This lets you import an authorization file to or export it directly from the authorization database.

4.5.2 Importing and Exporting in the Authorization Setup Program

On the main screen in the Authorization Setup Program, you will see the **Import** and **Export** buttons.

- Use the **Import** button to import an authorization file. The information in the authorization file will then be displayed, just as if you had configured it within the program.

- Use the **Export** button to export the information in the program to an authorization file. You are asked to specify a name and directory for your authorization file.

4.5.3 Using the Import/Export Utility

You can run the Import/Export utility either from a batch file or from the command line. As well as using the utility to import or export, you can use it to parse (check) your authorization file.

The name of the utility is *drive:\directory\bin\auth_imp_exp.exe* where *drive:\directory* is the drive and directory used for the Call Processing software installation. By default, this is *C:\Program Files\Intel\Ctc*.

4.5.3.1 Running the Import/Export Utility

The formats of the commands used to run the Import/Export Utility are shown in the following table:

To do this...	Use this command format...
Import an authorization file	<code>AUTH_IMP_EXP.EXE IMPORT import-file.txt [LOG=logfile.txt] [MINIMISE]</code>
Export an authorization file	<code>AUTH_IMP_EXP.EXE EXPORT export-file.txt [LOG=logfile.txt] [MINIMISE]</code>
Check the syntax of an authorization file	<code>AUTH_IMP_EXP.EXE PARSE import-file.txt [LOG=logfile.txt] [MINIMISE]</code>

where:

<i>import-file.txt</i>	is the name of the authorization file to be imported.
<i>export-file.txt</i>	is the name of the exported authorization file.
<i>logfile.txt</i>	is the name of the log file used to hold errors generated during the import or export. This is optional.
MINIMISE	specifies that the log file will not include errors generated by any of the Remove keywords (see Table 4-3). For example, it would not show errors generated if the authorization file contains a command to remove a user who is not actually in the authorization database. This is optional.

Example Import Command

In a batch file:

```
AUTH_IMP_EXP.EXE IMPORT Authusers_2_2002.txt MINIMISE
```

Example Export Command

From the command line:

```
C:\AUTH_IMP_EXP.EXE EXPORT Authusers.txt
```

4.5.4 Authorization File Format

In the authorization file, each user is defined by a group of keyword commands. You use these commands to:

- Add or remove users and passwords
- Add, remove or modify the devices a user is authorized to access
- Add or remove the client systems from which a user is allowed access

Each group of keyword commands must start with the command **User:username** and end with the command **EndUser:username**. The exception is the keyword **RemoveAllUsers**, which deletes all authorized users.

Table 4–3 shows the format of the keyword commands. Variables are in italics. Refer to Section 4.5.5 for an example authorization file.

Table 4–3 Authorization File Keyword Commands

Keyword	Description
User: <i>username</i>	Begins a group of commands. <i>username</i> is defined in Section 4.2.1.
EndUser: <i>username</i>	Ends a group of commands.
AddUser	Adds the user specified by the User: command.
RemoveUser	Removes the user specified by the User: command and all authorization information for that user.
AddPassword: <i>password</i>	Adds an encrypted password. Do not enter this command. Refer to Section 4.5.4.1 for more information.
AddNewPassword: <i>password</i>	Adds a clear text password. <i>password</i> is defined in Section 4.2.2. Also, see Section 4.5.4.1.
RemovePassword: <i>password</i>	Removes the specified password.
AddDevice: <i>logid:device-type:device-number:level</i>	Adds a device that the user is allowed to access: <ul style="list-style-type: none">• <i>logid</i> is the logical identifier of the link.• <i>device-type</i> is a device type; see Table 4–1, column 1.• <i>device-number</i> is a device number; see Table 4–2, column 1.• <i>level</i> is either Monitor or Full; see Section 4.2.3.5.

Table 4–3 Authorization File Keyword Commands (Continued)

Keyword	Description
<code>RemoveDevice:logid:device-type:device-number</code>	Removes the specified device.
<code>AddSystem:client-system</code>	Adds a client system from which the user is allowed access. <i>client-system</i> is defined in Section 4.2.4.
<code>RemoveSystem:client-system</code>	Removes a client system.
<code>RemoveAllDevices</code>	Removes all devices for a particular user.
<code>RemoveAllSystems</code>	Removes all client systems for a particular user.
<code>RemoveAllUsers</code>	Removes all users from the authorization database.

4.5.4.1 Defining Passwords

Use the **AddNewPassword:** command to enter a clear text password in the authorization file.

If you export an authorization file, you may see the **AddPassword:** command in the exported file. This command is used only for passwords that have been encrypted by the Call Processing Software. Because it is a security risk to export passwords as clear text, passwords in the authorization file are exported in encrypted form.

4.5.4.2 Multiple Device Numbers in the Authorization File

In the authorization file, you can use one wild card in a device number to specify multiple device numbers, just as you do in the Authorization Setup Program. However, you cannot specify a range in the authorization file (for example, 201 to 322). You must enter each number in the range separately.

The only supported wild card is the question mark (?).

4.5.4.3 Multiple Devices and Client Systems for a Single User

You can enter any number of devices and client systems for each user.

4.5.4.4 Command Execution

The commands in the authorization file are executed in the order in which they appear in the file.

4.5.4.5 RemoveUser Commands in Exported Authorization Files

When you export an authorization file, the software places a `RemoveUser:` command before each `User:` command. This is done so that when the file is imported, all authorization data for any existing user with that name will be

removed. The reason for this is to prevent obsolete user information remaining in the authorization database.

4.5.5 Example Authorization File

The first column in the table shows a simple authorization file with two users. The second column explains the commands.

Authorization File Commands	Explanation
User:AndreaAlison	Starts authorization information for user AndreaAlison.
RemoveUser	Removes any existing user with this name.
AddUser	Adds user AndreaAlison.
AddDevice:CSLondon:Agent:234:Full	Adds two devices on logical link CSLondon. One has the Agent Id 234; the other specifies all DN numbers starting with 23214. AndreaAlison has full access to all these devices.
AddDevice:CSLondon:Dn:23214?:Full	
AddSystem:AndreaDesk1	Adds three client systems that AndreaAlison is entitled to use to access the specified devices.
AddSystem:GeneralDesk	
AddSystem:GeneralDesk3	
EndUser:AndreaAlison	Ends authorization information for user AndreaAlison.
User:DavidDavidson	Starts authorization information for user DavidDavidson
RemoveUser	Removes any existing user with this name.
AddUser	Adds user DavidDavidson.
AddNewPassword:blueberry12fields	Adds a password in clear text. It is encrypted when stored in the database.
AddDevice:CSLondon:Agent:495:Full	Adds seven devices on logical link CSLondon. Six have Agent IDs in the range 495 to 500. The user has full access to these devices. One is a DN device with a Dn of 902348. The user has only monitor access to this device.
AddDevice:CSLondon:Agent:496:Full	
AddDevice:CSLondon:Agent:497:Full	
AddDevice:CSLondon:Agent:498:Full	
AddDevice:CSLondon:Agent:499:Full	
AddDevice:CSLondon:Agent:500:Full	
AddDevice:CSLondon:Dn:902348:Monitor	
or	

```
AddSystem: Davids-Desk  
AddSystem: General-Desk1  
  
EndUser: DavidDavidson
```

Adds two client systems that user DavidDavidson is entitled to use to access the specified devices.

Ends authorization information for user DavidDavidson.

Configuring Distributed Application Data

5.1 Introduction

This chapter describes how to configure the distributed data feature. This feature enables application data associated with a call to pass between call processing servers as the call passes between switches, in the case where the switches do not support the transfer of application data.

Note that this feature is not supported for all Call Processing Software licenses. Check your customer letter to see whether your license supports this feature.

5.1.1 Background

A call processing application can associate application data with a call. For example, it can associate a customer account number with a call, so that the number will remain associated with the call, even if the call is transferred to another agent.

If the call needs to be transferred to another switch, the application data should be transferred with the call. However, not all switches support the transfer of application data between switches.

The purpose of the distributed data feature is to enable application data to be transferred between call processing servers connected to different switches. When a client requests a call transfer to a remote switch, the application data is first stored on the call processing server on the network where the call was received. When the call is actually transferred, the application data is made available to the remote call processing server. The remote server then passes on the application data to the client receiving the transferred call.

5.1.2 Configuring Distributed Data

You use the Distributed Data Setup Program to configure local and remote links for distributed data. You can configure:

- Multiple call processing servers, each with at least one link to a switch.
- A single call processing server, with links to more than one switch.

Note: You must run the Distributed Data Setup Program on **each** call processing server that will store or receive distributed data.

5.1.2.1 Basic Configuration Method

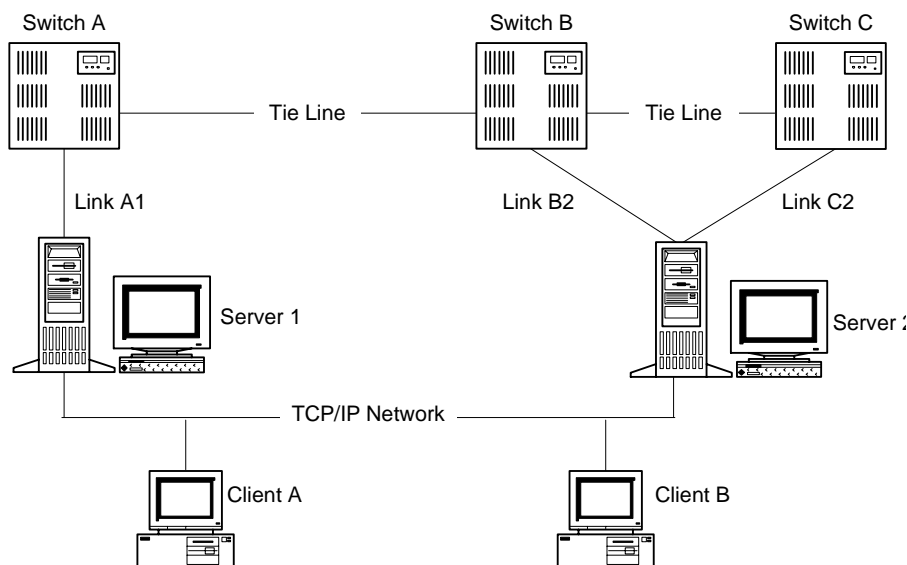
When you run the program, you configure pairs of links to be used for distributed data. For each pair, you identify:

- The server on which you are running the program. This is the **local server**.
- One or more **local links** that will handle distributed data.
- A **remote server**.
- One or more **remote links** that will handle distributed data.

5.1.2.2 Example

In Figure 5–1, Server 1 has a link to Switch A. Server 2 has two links; one to Switch B and one to Switch C. You want all links to be able to send distributed data to and receive distributed data from any other links.

Figure 5–1 Example Distributed Data Network



- On Server 1, configure an association between link A1 and each of the links on Server 2.

- On Server 2, configure an association between link B2 and link A1, link C2 and link A1, and also between link B2 and link C2.

5.1.3 How Data is Distributed

If you have configured links for distributed data, as shown in Figure 5–1, this is an example of what might happen when a call arrives at Client A's phone:

1. A call arrives at Client A's phone. The agent answers a call, and takes application data from the caller; for example, an account number.
2. The client application associates the application data with the call. This causes the data to be stored on Server 1.
3. The agent transfers the call, via Switch B, to Client B's phone.
4. The application data is made available to Server 2. There are two methods that can be used for this: either Server 2 requests the data from Server 1, or Server 1 sends the data when requested by a client application before the call is transferred.

The method that is used depends on your client application. Refer to the programming documentation for details.

5.2 Configuring the Distributed Data Network

This section describes how to use the Distributed Data Setup Program to configure distributed data.

5.2.1 Distributed Data Requirements

In order for the distributed data feature to work:

- The remote and local servers must be able to communicate with each other over TCP/IP.
- You must run the Distributed Data Setup Program on each server that will handle distributed data, as described in Sections 5.2.4 and 5.2.5.
- The client applications on the distributed data network must use the appropriate routines or methods to associate application data with calls transferred to a remote switch. For more information, refer to the programming documentation.

In addition to these requirements, please note that distributed data can normally only be used when each switch pair is connected by a tie trunk. A tie trunk is a telephone line that directly connects two switches or PBXs.

5.2.2 Before You Begin

Before you use the Distributed Data Setup program, you must do the following:

- Use the Configuration Program to configure a link between each server that will handle distributed data, and its associated switch. Refer to Section 3.2 for information about configuring links.
- In the Configuration Program, enable application data on each link. Refer to Table 3–1, page 3-8.

5.2.3 Required Information

Table 5–1 describes the information you will need when using the Distributed Data Setup Program:

Table 5–1 Distributed Data Setup Information Required

Information	Description
Local Links	The program displays a list of the logical identifiers configured on the local server. Select a link to configure for distributed data.
Remote Links	
Local or Remote	Select Local if the link to be configured is on the <i>same</i> server as the link selected under Local Links. Select Remote if the link to be configured is on a <i>different</i> server from the link selected under Local Links.
Server	If you selected Remote, type in the name of the remote server. If you selected Local, the Server field will automatically display the name of the local server.
Links	Type in the logical identifier of the link between the server and the switch to which it is connected.

Table 5–1 Distributed Data Setup Information Required (Continued)

Information	Description
Transport	<p>Select the transport interface for the connection between the local and the remote server. To use DCE/RPC, you must have selected it as a supported interface during installation or configuration. To use Java RMI or HTTP, you must have selected Java RMI as a supported interface during installation or configuration.</p> <ul style="list-style-type: none">• DCE/RPC Enables communication through the DCE/RPC interface. You must use DCE/RPC if the local link and the remote link are on the same server.• Java RMI Enables communication through the Java RMI interface.• HTTP Enables communication through an HTTP (Web) interface. Refer to Chapter 6 for a more detailed explanation. <p>See Section 2.6, page 2-9 for more information about interfaces.</p>
URL	<p>If you selected HTTP, enter the URL of the remote Web server used to reach the call processing server on the remote network. Refer to Chapter 6 for details</p>
Supply the following information for both the local link and the remote link	
Prefix (Area Code) for the switch.	<p>The prefix digits precede the switch device DN as can be seen in the Calling Line Identification (CLID) or Automatic Number Identification (ANI). For example, the Prefix may be the area code for the switch.</p>
Device No. Length	<p>The number of digits that make up the device number on the switch.</p>
Network ID	<p>The network identifier of the switch. Enter one of the following:</p> <ul style="list-style-type: none">• A network identifier of 0. Use this when both the local switch and the remote switch are either of the following:<ul style="list-style-type: none">– Different types of switch– CSTA switchesBoth switches must provide CLID.• A unique network identifier when both the local switch and the remote switch are of the same type (Nortel Meridian 1 and Avaya DEFINITY only). <p>Obtain the network identifier from the switch administrator who configured the identifiers on the switches.</p>

Table 5–1 Distributed Data Setup Information Required (Continued)

Information	Description
Supply the following information for the remote link only	
Switch Type	Select the switch type of the remote server's switch. Three switch types are supported: <ul style="list-style-type: none">• CSTA. This covers CSTA I, II and III.• DEFINITY.• Meridian 1.
Access No. Length	The number of digits that make up the access code required to reach the remote switch.

5.2.4 Starting the Distributed Data Setup Program

On the Start menu on the call processing server, select:

Programs → Intel NetMerge Call Processing Server → Distributed Data Setup Program

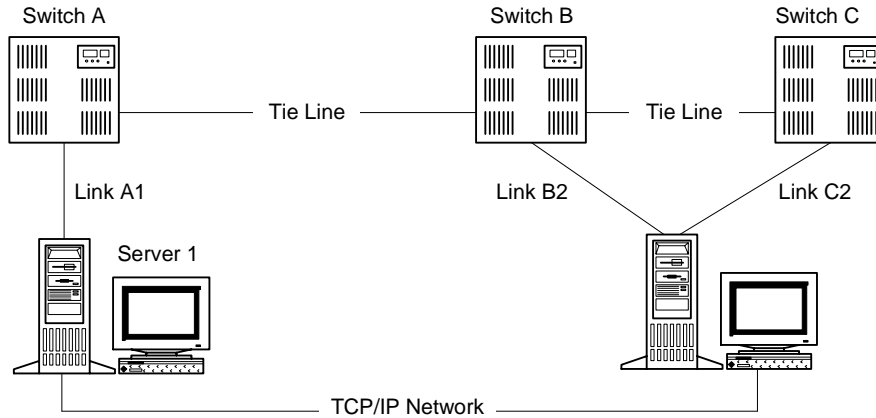
5.2.5 General Configuration Method

1. Start the Distributed Data Setup Program, as described in Section 5.2.4.
2. On the first screen, select a local link.
3. Enter the names of the remote server and link that will send distributed data to and/or receive it from the local link. Click the **OK** button.
4. On the second screen, enter information about the local and remote switches, and the transport between the local and remote links. Click the **OK** button. Refer to Table 5–1 for a list of the information you can enter.
5. You return to the first screen. If you have configured all the local/remote link pairs, click the **Exit** button.

5.3 Example Configuration

In Figure 5–2, Server 1 has a link to Switch A. Server 2 has two links, one to Switch B and one to Switch C.

Figure 5–2 Distributed Data Configuration



5.3.1 Configure Server 1

Follow these steps on Server 1:

1. Start the Distributed Data Setup Program.
2. Under Local Links, select the logical identifier, A1.
3. Under Remote Links, click the **Remote** button.
4. Type in Server 2 under Server, and B2 under Links.
5. Click the **OK** button.
6. On the next screen enter the required information under Local Link and Remote Link. The information under Local Link refers to Switch A; the information under Remote Link refers to Switch B. Refer to Table 5–1.
7. Click the **OK** button. This takes you back to the first screen.
8. Under Local Links, again select the logical identifier A1.
9. Under Remote Links, click the **Remote** button.
10. Again, enter Server 2 under Server. Then, type in C2 under Links.
11. Repeat steps 5 to 7. Then click **Exit**.

5.3.2 Configure Server 2

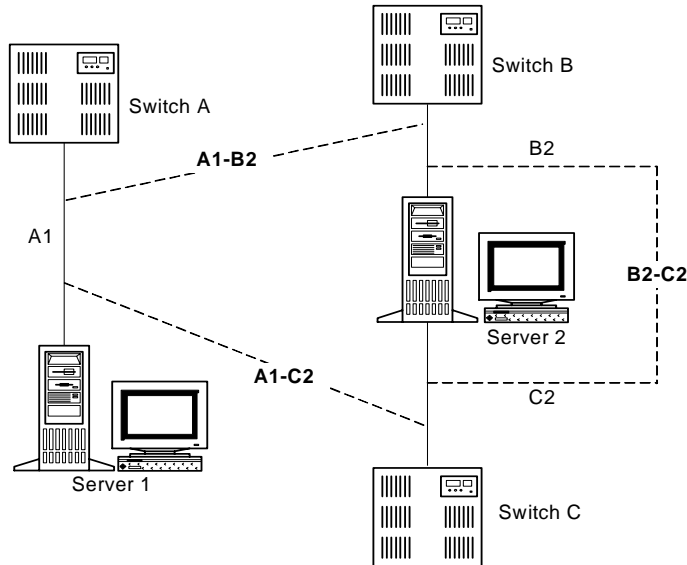
Follow these steps on Server 2:

1. Start the Distributed Data Setup Program, as described in Section 5.2.4.
2. Under Local Links, select the logical identifier, B2.
3. Under Remote Links, click the **Local** button. The name of the local server (Server 2) is displayed.
4. Select link C2 under Links.
5. Click the **OK** button.
6. On the next screen, enter the required information under Local Link and Remote Link. The information under Local Link refers Switch B; the information under Remote Link refers to Switch A. Refer to Table 5-1.
7. Click the **OK** button. This takes you back to the first screen.
8. Under Local Links, again select the logical identifier B2.
9. Under Remote Links, click the **Remote** button.
10. Enter Server 1 under Server. Then, type in A1 under Links.
11. Repeat steps 5 to 7.
12. Under Local Links, select the logical identifier C2.
13. Under Remote Links, click the **Remote** button.
14. Enter Server 1 under Server. Then, type in A1 under Links.
15. Repeat steps 5 to 7. Then click **Exit**.

5.3.3 Result of Configuration

You have now configured link pairs A1-B2 and A1-C2 on Servers 1 and 2, and link pair B2-C2 on Server 2, as shown in Figure 5-3.

Figure 5–3 Link Pairs for Server 1 and Server 2



5.4 Modifying Link Information

To modify information for a link pair, follow these steps in the Distributed Data Setup Program:

1. Select the local link that you want to modify.
2. Select the remote server and remote link that you want to modify. Click the **OK** button.
3. On the next screen, modify the information as desired.
4. Click the **OK** button.

5.5 Deleting Link Information

To delete a link pair, follow these steps in the Distributed Data Setup Program:

1. Select the local link that you want to delete.
2. Select the remote server and remote link. Click the **OK** button.
3. On the next screen, click the Delete button.

Configuring Web Access Using SOAP

6.1 Introduction

The Call Processing Software can, working with a Web server, use the Simple Object Access Protocol (SOAP) to allow Call Processing Software messages to pass through a firewall. This chapter gives an overview of this process, and describes the configuration steps and programming tasks required to make it work.

6.1.1 What is SOAP?

SOAP is a protocol that enables programs to make remote procedure calls (RPC) using XML. SOAP typically uses HTTP as its transport. SOAP is a World Wide Web Consortium (W3C) working draft.

SOAP uses XML for invoking methods on servers, services, components and objects. The SOAP specification specifies a number of HTTP headers that facilitate firewall/proxy filtering. It also specifies an XML vocabulary that is used for representing method parameters, return values and exceptions.

The Call Processing Software installation procedure automatically installs the Apache SOAP implementation on the call processing server. Apache SOAP is an open-source implementation of SOAP V1.1.

6.1.2 Using the Jakarta Tomcat Servlet Container

Apache SOAP expects to be hosted by a *servlet container*. A servlet container is a Java application that manages and invokes *servlets* on behalf of users. A servlet is a Java class that can be invoked and executed on a server to respond to a request from a client.

In the case of SOAP, the servlet container is required to process SOAP requests and send SOAP responses. (With Apache SOAP, this is accomplished by the Apache SOAP `rpcrouter` servlet.)

The Call Processing Software installation procedure automatically installs the Jakarta Tomcat servlet container on the call processing server. Tomcat is the

servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies.

Tomcat can act as a Web server. However, it can also be integrated with existing Web servers.

6.1.3 Call Processing Software SOAP Configurations

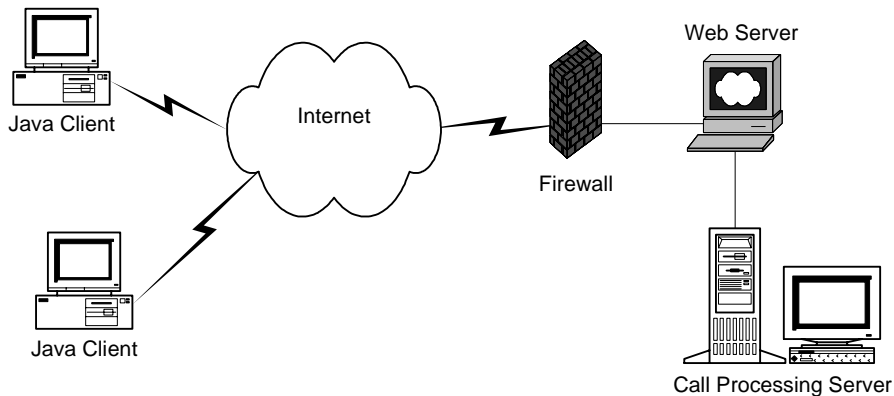
There are two possible Call Processing Software configurations that use SOAP:

- Call processing Java clients use the Java call processing API to communicate through a firewall with your call processing server.
- Call processing servers exchange distributed data through IP firewalls, as shown in Figure 6–2. Distributed data is described in Chapter 5.

6.1.4 Overview: Java Client Access

In Figure 6–1, the call processing server and the Web server are on an intranet protected by a firewall. The Java clients are outside the firewall. The Java clients can exchange messages with the call processing server through the Web server.

Figure 6–1 Client Access Through an IP Firewall



In this configuration, the required files for running SOAP have been copied to the Java client. In addition, Tomcat has been integrated with the Web server. Finally, the Java client application supplies the URL and port number of the Web server when it creates a channel (see Section 6.5.2).

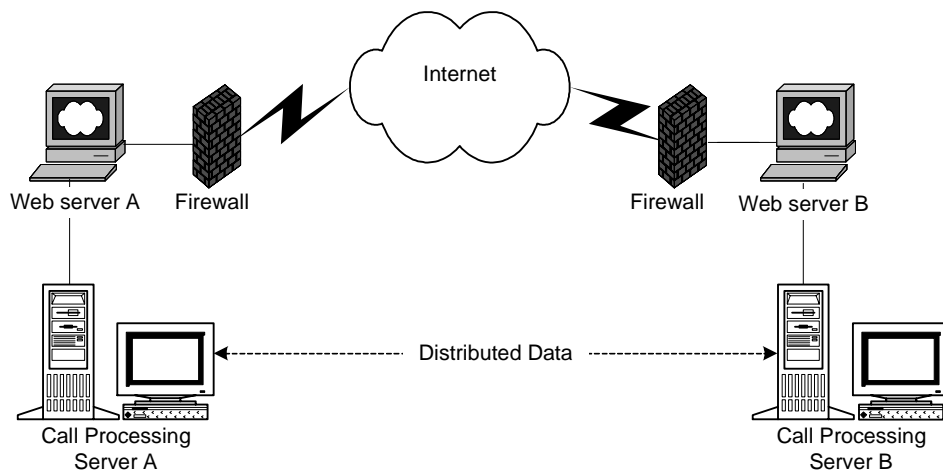
The presence of a URL and port number in the `ctcChannel` object tells the Call Processing Software that it needs to use SOAP. The SOAP software on the client then translates the API calls to text-based SOAP messages

In this example, the SOAP messages traverse the Internet, pass through the firewall, and are sent to the Web server., which passes them to Tomcat on the call processing server. Finally, the SOAP messages are converted back to their original format so that they can be processed by the call processing server.

6.1.5 Overview: Distributed Data Servers

In Figure 6–2, call processing server A and Web server A are on intranet A, protected by a firewall, while call processing server B and Web server B are on intranet B, protected by a firewall. The servers are configured to exchange distributed application data over the Internet via the Web servers. Refer to Chapter 5 for a complete description of the distributed data feature.

Figure 6–2 Distributed Data Access Through an IP Firewall



In this configuration, Tomcat on call processing server A has been integrated with Web server A, while Tomcat on call processing server B has been integrated with Web server B. In addition, when the distributed data feature was configured on the call processing servers, HTTP was specified as the transport and the URL of the remote Web server specified as the URL (see Table 5–1).

Before the application data is sent from a call processing server, it is translated into SOAP format. The SOAP messages traverse the Internet, using HTTP transport. They are then passed by the remote firewall, and sent to the Web

server, which passes them to Tomcat. Finally, the SOAP messages are converted to their original format so that they can be processed by the call processing server.

6.2 Installation

During the call processing server installation procedure, you are asked if you wish to install SOAP Access Components. These are the components used to implement the various SOAP services, including Tomcat. You are asked this because installing Tomcat could compromise the security of your call processing server, as Tomcat is a Web server.

If you choose to install the SOAP Access Components, the Call Processing Software automatically does the following:

- Installs the software required to run SOAP on the call processing server. This includes both the call processing API SOAP service and the distributed data SOAP service.
- Starts Tomcat as a Windows service called **Jakarta Tomcat 4.0.3**. This service provides the standard Tomcat features and services, including Web server implementation. Tomcat is configured via the `server.xml` file, as described in the Tomcat documentation.

Table 6–1 lists additional tasks you may need to complete.

Table 6–1 SOAP Tasks

Task	When it is required	Refer to Section...
Integrate Tomcat with your Web server	If you do not wish to use Tomcat as your usual Web server.	6.3
Configure SOAP to use log files	If you want to log SOAP call processing API messages.	6.4
Install SOAP files on the external Java client	To pass SOAP messages between Java clients and the call processing server.	6.5
Configure the distributed data feature to use SOAP	To pass SOAP messages between distributed data call processing servers.	6.6

6.2.1 Product Versions Installed

Table 6–2 lists the main third-party software installed for SOAP, with version number and location on the call processing server. It is strongly recommended that you use these versions and implementations.

In the table, *drive:\directory* is the installation drive and directory. This is C:\Program Files\Intel\Ctc by default. For Jakarta Tomcat, *drive:* is the installation drive. This is C: by default.

Table 6–2 Software Installed to Implement SOAP

Software	Version	Location
Apache SOAP	2.2	<i>drive:\directory\java</i>
Jakarta Tomcat (Version 4 is also known as Catalina.)	4.0.3	<i>drive:\ctcjava\jakarta-tomcat-4.0.3</i>
Apache Xerces (XML parser)	1.4.1	<i>drive:\directory\java</i>
JavaMail	1.2	<i>drive:\directory\java</i>
Javabeans Activation Framework	1.0.1	<i>drive:\directory\java</i>

6.3 Integrating Tomcat With Your Web Server

Although Tomcat is itself a Web server, you may prefer to use an existing Web server, and only use Tomcat as a Web application server. In this case, you need to integrate Tomcat with your existing Web server.

The Tomcat documentation contains instructions for integrating Tomcat with the following Web servers:

- Apache Web server
- IIS Web server

Integrating Tomcat with a Web server means that you need to set up the Web server to forward servlet requests to Tomcat, and set up Tomcat to respond to requests from the Web server.

6.3.1 Integrating Tomcat with an Apache Web Server

Follow these steps:

1. Open the following file on your call processing server

drive:\Ctcjava\jakarta-tomcat-4.0.3\webapps\tomcat-docs\index.html

2. On the Web page, under **Administrators**, click **Server Configuration Reference**.
3. Under **Overview**, click **AJP**.
4. The instructions for integrating Tomcat with an Apache Web server are under **Nested Components/Configuration HOWTOs Apache**.

6.3.2 Integrating Tomcat with an IIS Web Server

Follow these steps:

1. Follow steps 1 to 3 in Section 6.3.1.
2. The instructions for integrating Tomcat with an IIS Web server are under **Nested Components/Configuration HOWTOs IIS 4.x and 5.x**.

Please refer to the Call Processing Software release notes for any updates to this procedure.

6.4 Configuring SOAP to Use Log Files

By default, there is no logging of call processing API requests and responses, or distributed data messages in SOAP format. This section describes how to enable and disable logging.

6.4.1 Logging Call Processing API Messages

To start the logging of call processing API messages in SOAP format, follow these steps:

1. To stop the deployment of the current call processing API SOAP service, run the following batch file:

```
drive\directory\java\ctcapi_undeploy.bat
```

2. To start logging, run the following batch file:

```
drive\directory\java\ctcapi_deploy_logging.bat
```

To turn logging off again, first repeat step 1. Then, run the following batch file:

```
drive\directory\java\ctcapi_deploy.bat
```

6.4.2 Logging Distributed Data Messages

To start the logging of distributed data messages in SOAP format, follow these steps:

1. To stop the deployment of the current distributed data SOAP service, run the following batch file:

```
drive\directory\java\ctcdistdata_undeploy.bat
```

2. To start logging, run the following batch file:

```
drive\directory\java\ctcdistdata_deploy_logging.bat
```

To turn logging off again, first repeat step 1. Then, run the following batch file:

```
drive\directory\java\ctcdistdata_deploy.bat
```

6.4.3 Location of Log Files

All log files are in the following location:

```
drive:\Ctcjava\jakarta-tomcat-4.0.3\logs
```

The data of each log is included in its file name. After a log file is started for a date, all logging information for that date is appended to that log file.

6.5 Using Soap on an External Java Client

To use SOAP on an external Java client, you need to do the following:

- Install the required software.
- Use the required programming arguments in your client application.

6.5.1 Installing the Required Software

Refer to Chapter 13 for information about installing the call processing API for Java clients. Table 13–2 lists the SOAP files you need to copy.

6.5.2 Programming Arguments

In the client application, you must include the following parameters in the constructor for creating a `ctcChannel` object:

- `httpURL`. This is the URL of the Web server used to reach the call processing server.
- `httpPort`. This is the port number used to reach the Web server. The default is 80.

6.6 Using Soap on a Distributed Data Server

To use SOAP on a distributed data server, you need to supply the required parameters in the Distributed Data Setup Program. These are:

- Under Transport, on the second screen, select HTTP.
- In the URL box, enter the URL of the Web server used to reach the call processing server.

6.7 Supported Methods for Java Clients

Note that only a subset of Java call processing API methods are supported for use with SOAP. Refer to the Release Notes for a list of supported methods.

Starting and Stopping the Call Processing Server

This chapter describes how to start the call processing server software and how to start configured links between the server and a switch.

7.1 Starting the Server Software

This section describes how to start the server software.

7.1.1 Automatic Startup

The server software is configured to start up automatically when the system boots up. This means that a Windows service is started each time the system boots up.

To start the software, and any links configured to start up automatically with the software, reboot the server.

Events Logged

When the server and links have started successfully, the following events are logged:

- ctcServerStarted
- ctcInterfaceStarted
- ctcLinkUp

Depending on your Call Processing Software license, you may also see the ctcLinkEnabled event.

If you have multiple links configured, the log will show multiple instances of the same events.

For more information about event logging, refer to the online Problem Solving and Monitoring Help. From the Start menu on the server, select:

- Programs → Intel NetMerge Call Processing Server → Problem Solving and Monitoring Help

7.1.2 Starting the Server Software Manually

This section describes how to start the server software manually. Follow the appropriate procedure for your operating system.

7.1.2.1 Windows NT

To start the CtcServer service in Windows NT, follow these steps:

1. From the Control Panel, open Services.
2. Select **CtcServer** from the list of available Services.
3. Click on the **Start** button.

7.1.2.2 Windows 2000

To start the CtcServer service in Windows 2000, follow these steps:

1. In the Control Panel, select the Administrative Tools icon.
2. Select the Services icon.
3. Select **CtcServer** from the list of available Services.
4. On the Action menu, select Start. Alternatively, click on the start (right-arrow) button on the toolbar.

7.1.2.3 Windows XP

To start the CtcServer service in Windows XP, follow these steps.

1. In the Control Panel, select Performance and Maintenance.
2. Select the Administrative Tools icon.
3. Select Services.
4. Select **CtcServer** from the list of available Services.
5. On the Action menu, select Start. Alternatively, click on the start (right-arrow) button on the toolbar.

7.1.2.4 Events Logged

When the server and link have started successfully, the following events are logged:

```
ctcServerStarted  
ctcInterfaceStarted  
ctcLinkUp
```

For more information about event logging, refer to the online Problem Solving and Monitoring Help. From the Start menu on the server, select:

Programs → Intel NetMerge Call Processing Server → Problem Solving and Monitoring Help

7.2 Stopping the Server Software

This section describes how to stop the call processing server software. Select the appropriate procedure for your operating system.

7.2.1 Windows NT

To stop the CtcServer service in Windows NT, follow these steps:

1. Select the Services icon in the Windows Control Panel.
2. Select **CtcServer** from the list of available Services.
3. Click on the **Stop** button.
4. Click on the **OK** button.

7.2.2 Windows 2000

To stop the CtcServer service in Windows 2000, follow these steps:

1. Select the Administrative Tools icon in the Windows Control Panel.
2. Select the Services icon.
3. Select **CtcServer** from the list of available Services.
4. On the Action menu, select **Stop**.

7.2.3 Windows XP

To stop the CtcServer service in Windows XP, follow these steps.

1. In the Control Panel, select Performance and Maintenance.
2. Select the Administrative Tools icon.
3. Select Services.
4. Select **CtcServer** from the list of available Services.
5. On the Action menu, select Stop.

7.2.4 Events Logged

If the link is up when you stop the server software, the `ctcLinkDown` and `ctcServerStopped` events are logged.

7.3 Starting a Link Between the Server and the Switch

The link must be started for messages to pass between the server and the switch. This section describes how to start a link.

7.3.1 Starting the Link Automatically

If you enabled Auto Start Link in the Configuration Program, the link is started automatically when the server software is started. When the server software starts, it uses the information held in the system registry to enable configured links. See Chapter 3 for information about the Configuration Program and the Auto Start Link option.

7.3.2 Starting the Link Manually

If a link is not configured to start when the server software starts, you will need to start it with the Control Program before it can be used. Follow these steps:

1. Start the Control Program. From the Start menu on the server, select:
Programs → Intel NetMerge Call Processing Server → Control Program.
2. On the screen, you will see a list of logical identifiers for configured links. Select the logical identifier of the link you want to start.
3. Click the **On** button. Note that if the link is already on, the button will say **Off**.

7.3.3 Events Logged

The `ctcServerStarted` event is logged when the server software is successfully started, and the `ctcLinkUp` event is logged when a link is successfully started.

For more information about event logging, refer to the online Problem Solving and Monitoring Help. From the Start menu on the server, select:

Programs → Intel NetMerge Call Processing Server → Problem Solving and Monitoring Help

7.3.4 Errors at Link Startup

If you cannot start the link to the switch, you can use the Control Program to obtain more information. If this does not isolate the problem, refer to the

problem solving procedures in the online Problem Solving and Monitoring Help.
From the Start menu on the server, select:

Programs → Intel NetMerge Call Processing Server → Problem Solving
and Monitoring Help

Part II

Installing the Call Processing API Software

Each of the chapters in Part II describes how to install the call processing API software on a supported client platform, and check communication with the call processing server and the switch.

The following table lists each supported platform, and the chapter that describes it.

For this type of client...	Follow the procedure in...
Distributed Computing Environment (DCE) Based	
HP-UX	Chapter 8
OpenVMS	Chapter 9
Solaris	Chapter 10
Tru64 UNIX	Chapter 11
Windows systems. Supported systems are: Windows 9x, Windows NT, Windows 2000 and Windows XP	Chapter 12
Java Remote Method Invocation (RMI) Based	
Java Runtime Environment	Chapter 13

Installing the Call Processing API on HP-UX

8.1 Overview

The following table provides an overview of the contents of this chapter.

For details of...	See Section...
Hardware requirements	8.2
Software requirements	8.3
Installation instructions	8.4
Files installed	8.5
What to do after installation	8.6
Checking communication	8.7
Removing the software	8.8

Note that the management API is installed as part of the call processing API software kit. For more information, refer to Section 1.2.3.

8.2 Hardware Requirements

To install the call processing API on an HP-UX system, you require:

- One of the following HP-UX systems in the same network as the call processing server:
 - A 700-series system (PA_RISC_1.0 or PA_RISC_1.1 architecture)
 - An 800-series system (PA_RISC_1.0 architecture)

For more details of the hardware required, contact Hewlett-Packard† or your hardware supplier.

- 1.5 Mbytes of free hard disk space to load the software.

These are minimum requirements for installing and using the call processing API and the call processing management API on an HP-UX system. To run

additional software, you may need more resources, for example, additional disk space and memory.

8.3 Software Requirements

To install the call processing API on an HP-UX system, you require:

- HP-UX Version 10.2 or 11.0.
- If you are developing call processing applications on your HP-UX system, you also require HP DCE/9000 Application Development Tools.

For more information, contact Hewlett-Packard.

Note that the network protocol TCP/IP is provided as part of HP-UX. Check that the call processing server is configured to communicate with your HP-UX system using TCP/IP (see Section 3.3).

8.3.1 Defining the Call Processing Server Name and Address

To ensure that the call processing server and your HP-UX system can communicate, the name and network address of the call processing server must be defined either in your network server database (if your network has one) or on your HP-UX system. For example, for a TCP/IP network, you can add the call processing server TCP/IP name and address to the hosts file.

For more information, refer to your network management documentation or see your network manager.

8.4 Installation Procedure

Follow these steps:

1. Log in as a superuser (login name `root`) on the local system.

For more information about superuser privileges, refer to your HP-UX system manager documentation.

2. Create a directory for the call processing API software kit and change to that directory. For example:

```
# mkdir /ctc_subset
# cd /ctc_subset
```

Both the call processing API software and the call processing management API software will be copied to this location as part of the same kit.

3. Copy the file `ctc600.HP` from the call processing server. For example, using FTP:

```
# ftp
ftp> open server_name
```

where *server_name* is the name of the server system. Enter *username* and *password* when prompted and then locate the directory used for the call processing server software installation. The HP-UX kit is located in the subdirectory `\clients\hpux`.

For example:

```
ftp> cd c:\Program Files\Intel\ctc\clients\hpux
```

Enter the following commands to copy the file:

```
ftp> binary
ftp> get ctc600.hp
ftp> quit
```

4. Unpack the kit from the tape archive file:

```
# tar -xvf ctc600.hp
```

5. Use the **swinstall** utility to install the call processing API. Enter:

```
# swinstall -s /ctc_subset
```

For more information on the `swinstall` utility, refer to your HP-UX system management documentation.

8.5 Files Installed

Table 8–1 describes the files installed during the call processing API installation procedure and their locations.

Table 8–1 Call Processing API Files Installed on HP-UX

Description	Location
Call processing definitions files to include in your application	<code>/usr/include/</code>
Executables for the Call processing utilities	<code>/usr/bin/</code>
Call processing API and management API runtime libraries	<code>/usr/shlib/</code>
Source code for the example application	<code>/opt/intel/ctc600/ctc/example</code>

8.6 What to Do Next

When the installation is complete:

1. If you did not read the release notes installed on the call processing server, read them now. See Section 2.8.1.
2. Check that your HP-UX system can communicate with the call processing server and the switch. See Section 8.7.

8.7 Checking Communication

You use the CTC Test program to check communication between your HP-UX call processing client, the call processing server, and the switch.

Before you start CTC Test, make sure that the communications link between the call processing server and the switch is available and ready for use. For more information, refer to Section 3.10.

For detailed information about CTC Test, refer to the *Intel NetMerge Call Processing Software CTC Test User's Guide* (provided as a PDF file).

8.7.1 Procedure

Follow these steps:

1. Gather the following information:

Item	Description
DN	A telephone number on the switch that you can use to check the link. You assign a channel to this DN when you run CTC Test.
Server Name	The name of the call processing server. For example, SKIDDAW.
Logical Identifier	The logical identifier of the link between the server and the switch. For example, SERV1.
Network protocol	A value used to identify the protocol used between the call processing server and the call processing client. Specify the value: ncacn_ip_tcp

2. Start CTC Test by entering the following command:

```
# ctc_test
```

The command prompt `ctcTest>` will appear.

3. Enter the **assign** command at the prompt, in the following format:

```
ctcTest> assign dn server_name logical_identifier ncacn_ip_tcp
```

Use the data you gathered in Step 1 for the command parameters. For example:

```
# ctcTest> assign 2345 skiddaw ctcserv1 ncacn_ip_tcp
```

This command assigns a channel to DN **2345**. The server name is **skiddaw** and the logical identifier for the link between the server and the switch is **ctcserv1**. The required value for the network transport is **ncacn_ip_tcp**.

8.7.2 Using CTC Test with Device Level Authorization

You need to use the **secassign** command instead of the **assign** command if both of the following are true:

- In the Configuration Program, you have enabled both Device Level Authorization, and either System or Application password checking. Refer to Chapter 3, Table 3-1.
- You have configured one or more authorized users, as described in Chapter 4.

Note: If you have enabled Device Level Authorization but have selected None for password checking, you should use the **assign** command rather than the **secassign** command.

8.7.2.1 Procedure

Follow these steps:

1. Gather the information listed in Section 8.7.1, Step 1. In addition, you must supply a valid user name and password for an authorized user. Refer to Sections 4.2.1 and 4.2.2 for details.
2. Start CTC Test as described in Section 8.7.1, Step 2.
3. Enter the **secassign** command at the prompt:

```
ctcTest> secassign dn server_name logical_identifier ncacn_ip_tcp
```
4. You are prompted for a user name. Enter the name of an authorized user.
5. You are prompted for a password. Enter the user's password.

8.7.3 What to Do If CTC Test Fails

If CTC Test returns a fail message, check that:

- All details you specified are correct. The specified telephone number must be a valid number on the switch.
- The call processing server software is installed and running on the server.
- The name for the call processing server is defined in your network name database or on your local call processing client system.
- If you are using secassign, check that the user name and password are configured in the authorization database, and are associated with the device, the link and the client that you specified.
- The link between the server and the switch is enabled.
- The network connection between the call processing client and the call processing server is up and running.

If the procedure still fails, start the Control Program on the server and click the **Version** button to check the version of the server software.

The server must run Call Processing Software V6.0 software for compatibility with clients running the API V3.0, V4.0, V5.0 or V6.0 software. Refer to Section 1.3 for information about upgrading the server software.

If the procedure still fails, contact Intel.

8.8 Removing the Software

To remove the call processing API and management API software from your system:

1. Log in as superuser (login name `root`).
2. Use the `swremove` utility to remove the call processing API software.

For more information about the `swremove` utility, refer to your HP-UX system management documentation.

Installing the Call Processing API on OpenVMS

9.1 Overview

The following table provides an overview of the contents of this chapter.

For details of...	See Section...
Hardware requirements	9.2
Software requirements	9.3
Installation instructions	9.4
Files installed	9.5
What to do after installation	9.6
Checking communication	9.7

Note that the management API is installed as part of the call processing API software kit. For more information, refer to Section 1.2.3.

9.2 Hardware Requirements

The following hardware is required:

- An OpenVMS system that can connect to a call processing server over TCP/IP. For more details of the hardware required, contact Compaq or your hardware supplier.
- 1 Mbyte of free space (2000 blocks) during the installation.
- 500 Kbytes of free space (1000 blocks) after installation.

These are minimum requirements for installing and using the call processing API and the call processing management API on an OpenVMS system. To run additional software, you may need more resources, for example, additional free space and memory.

9.3 Software Requirements

The following software is required:

- OpenVMS Alpha Version 7.1

The version that you use must include the following classes:

VMS Required Saveset

Network Support

Programming Support

- Compaq TCP/IP Services for OpenVMS for communication with the call processing server.
- Compaq Distributed Computing Environment (DCE) for OpenVMS V2.0 or later, to support the Remote Procedure Call (RPC) services.
- If you are writing a call processing program on the OpenVMS system, you also require the DCE Application Developer's Kit for OpenVMS.

9.3.1 DCE RPC Environment

The distributed call processing client/server uses DCE to support the RPC services, providing a common API across different platforms.

To run Call Processing Software V6.0, you require Compaq DCE for OpenVMS V2.0 or later running on your OpenVMS system. Note that when you install the Compaq DCE for OpenVMS software, you should select the following option from the Initial Configuration Menu:

2) RPC_only

This option enables you to install and run Call Processing Software V6.0. If you want to use the Compaq DCE for OpenVMS software with other software, refer to the *Compaq DCE for OpenVMS VAX and OpenVMS Alpha Installation and Configuration* guide for more information.

9.3.2 Privileges

Install the call processing API software from the system manager's account with your default device and directory set to SYSSUPDATE. Alternatively, ensure that you have the following OpenVMS privileges:

CMKRNL
EXQUOTA
NETMBX

SYSNAM
SYSPRV
TMPMBX

For more information about these privileges, refer to your OpenVMS documentation.

9.3.3 Defining the Call Processing Server Name and Address

To ensure that the call processing server and the OpenVMS client can communicate, the name and network address of the call processing server must be defined either in your network server database (if your network has one) or on your call processing client. For example, for a TCP/IP network, the call processing server's TCP/IP name and address must be added to the hosts file.

For more information, refer to your network management documentation or see your network manager.

9.4 Installation Procedure

To install the call processing API, follow these steps:

1. Log in to a privileged account:

```
Username: SYSTEM
```

```
Password:
```

2. The call processing API installation kit for OpenVMS is loaded onto the server when the call processing server software is installed. Create a directory to copy the installation files from the call processing server, and make that your default directory:

```
$ CREATE/DIRECTORY kit_directory
```

```
$ SET DEFAULT kit_directory
```

where *kit_directory* is the specification for the new directory. For example, DISK\$SYSTEM01:[KITS.CTC].

3. Copy the installation files from the call processing server using FTP:

```
$ FTP
```

```
FTP> OPEN server_name
```

where *server_name* is the name of the call processing server node.

Enter *username* and *password* when prompted and then enter the following commands (in lowercase) to copy the file:

```
ftp> cd drive:\directory\CLIENTS\VMS
ftp> ascii
ftp> get ctccvt.com
ftp> binary
ftp> get ctc060.a
ftp> get ctc060.b
ftp> quit
```

where *drive:/directory* is the drive and directory used for the call processing server software installation. For example:

```
C:\Program Files\Intel\Ctc
```

The file CTCCVT.COM must be copied as an ASCII text file, and the savesets must be copied as binary image files.

4. Convert the installation files to VMS saveset format. Enter:

```
$ @CTCCVT CTC
```

5. Set your default directory to SYSSUPDATE, and invoke VMSINSTAL:

```
$ SET DEFAULT SYSSUPDATE
```

```
$ @VMSINSTAL CTC060 device:directory
```

where *device:directory* identifies the directory that you created in step 2, (*device* is the name of the volume, and *directory* is the name of the directory). The VMSINSTAL command starts the installation procedure and displays the following question:

```
* Are you satisfied with the backup of your system disk [YES]?
```

If you are not satisfied, enter No, and the installation finishes. If you are satisfied, press Return, and the installation continues:

```
The following products will be processed:
```

```
CTC V6.0
```

```
Beginning installation of CTC V6.0 at hh:mm
```

```
%VMSINSTAL-I-RESTORE, Restoring product saveset A...
```

6. You have now supplied all the required information and can monitor the installation.

VMSINSTAL copies the installation files (see Section 9.5 for a list of the directories to which they are moved) and runs the Installation Verification Procedure (IVP). Section 9.4.1 contains guidance if the IVP failed during the installation.

9.4.1 The IVP Fails

The installation procedures for the call processing API software on OpenVMS contain an IVP which checks that the files have been correctly installed. If the IVP fails, take the following actions:

1. Check that the system quotas or resources on your system are adequate.
2. Check that the prerequisite software is installed.
3. Ensure that the hardware on your system meets the stated requirements.
4. Check that there are no problems with the system disk you are using.

If none of the above actions isolates the problem, contact Intel.

9.5 Files Installed

Table 9–1 describes the files installed during the call processing API installation procedure and their locations.

Table 9–1 Call Processing API Files Installed on OpenVMS

Description	Location
Call Processing Software definitions files and the call processing API and management API executables	SYSSLIBRARY
Executables for the Call Processing Software utilities	SYSSCOMMON:[SYS\$TEST.CTC]
Source code for the example application	SYS\$EXAMPLE
Command procedure for starting the call processing client	SYSSSTARTUP
Command procedure for starting the CTC Test program	SYS\$TEST

9.6 What to Do Next

When the installation is complete:

1. If you did not read the release notes installed on the call processing server, read them now. See Section 2.8.1.
2. Start the call processing API software. See Section 9.6.1
3. Check that your OpenVMS system can communicate with the call processing server and the switch. See Section 9.7.

9.6.1 Starting the Software

To start the call processing API software, run the startup file:

```
$ @SYS$STARTUP:CTC_STARTUP.COM
```

If you want the call processing API software to start whenever the system restarts, include the following command in the site-specific startup command file (SYSTARTUP_VMS.COM):

```
$ @SYS$STARTUP:CTC_STARTUP.COM
```

9.7 Checking Communication

You use the CTC Test Program to check communication between your OpenVMS call processing client, the call processing server, and the switch.

Before you start CTC Test, make sure that the communications link between the call processing server and the switch is available and ready for use. For more information, refer to Section 3.10.

For detailed information about CTC Test, refer to the *Intel NetMerge Call Processing Software CTC Test User's Guide* (provided as a PDF file).

9.7.1 Procedure

Follow these steps:

1. Gather the following information:

Item	Description
DN	A telephone number on the switch that you can use to check the link. You assign a channel to the DN when you run CTC Test.
Server Name	The name of the call processing server system. For example, SCAFELL.

Item	Description
Logical Identifier	The logical identifier of the link between the server and the switch. For example, SERV1.
Network Protocol	A value used to identify the protocol used between the call processing server and the client. Specify the value: ncacn_ip_tcp

2. Enter the following command:

```
$ RUN SYS$TEST:CTC_TEST
```

The command prompt `ctcTest>` will appear.

3. Enter the **assign** command at the prompt, in the following format:

```
ctcTest> assign dn server_name logical_identifier ncacn_ip_tcp
```

Use the data you gathered in Step 1 for the command parameters. For example:

```
# ctcTest> assign 2345 skiddaw ctcserv1 ncacn_ip_tcp
```

This command assigns a channel to DN **2345**. The server name is `skiddaw` and the logical identifier for the link between the server and the switch is `ctcserv1`. The required value for the network transport is `ncacn_ip_tcp`.

9.7.2 Using CTC Test with Device Level Authorization

You need to use the **secassign** command instead of the **assign** command if both of the following are true:

- In the Configuration Program, you have enabled both Device Level Authorization, and either System or Application password checking. Refer to Chapter 3, Table 3-1.
- You have configured one or more authorized users, as described in Chapter 4.

Note: If you have enabled Device Level Authorization but have selected None for password checking, you should use the `assign` command rather than the `secassign` command.

9.7.2.1 Procedure

Follow these steps:

1. Gather the information listed in Section 9.7.1, Step 1. In addition, you must supply a valid user name and password for an authorized user. Refer to Sections 4.2.1 and 4.2.2 for details.

2. Start CTC Test as described in Section 9.7.1, Step 2.
3. Enter the **secassign** command at the prompt:

```
ctcTest> secassign dn server_name logical_identifier ncacn_ip_tcp
```
4. You are prompted for a user name. Enter the name of an authorized user.
5. You are prompted for a password. Enter the user's password.

9.7.3 What to Do If CTC Test Fails

If CTC Test returns a fail message, check that:

- All details you specified are correct. The specified telephone number must be a valid number on the switch.
- The call processing server software is installed and running on the call processing server.
- The name for the call processing server is defined in your network name database or on your local call processing client system.
- If you are using secassign, check that the user name and password are configured in the authorization database, and are associated with the device, the link and the client that you specified.
- The link between the call processing server and the switch is enabled.
- The network connection between the call processing client and the call processing server is up and running.

If the procedure still fails, start the Control Program on the server and click the **Version** button to check the version of the server software.

The server must run Call Processing Software V6.0 software for compatibility with clients running the API V3.0, V4.0, V5.0 or V6.0 software. Refer to Section 1.3 for information about upgrading the server software.

If the procedure still fails, contact Intel.

Installing the Call Processing API on Solaris

10.1 Overview

The following table provides an overview of the contents of this chapter.

For details of...	Section...
Hardware requirements	10.2
Software requirements	10.3
Installation instructions	10.4
Files installed	10.5
What to do after installation	10.6
Checking communication	10.7
Removing the software	10.8
Reinstalling the software	10.9

Note that the management API is installed as part of the call processing API software kit. For more information, refer to Section 1.2.3.

10.2 Hardware Requirements

The following hardware is required:

- A SPARC[†] system suitable for installing Solaris software and in the same network as the call processing server. For more details of the hardware required, contact Sun or your hardware supplier.
- 1.5 Mbytes of free hard disk space to load the software.

These are minimum requirements for installing and using the call processing API and the call processing management API on a Solaris system. To run additional software, you may need more resources, for example, additional disk space and memory.

10.3 Software Requirements

The following software is required:

- Solaris Version 7 or 8.
- IBM† DCE Version 3.1 or 3.2 for Solaris Base Services software to support the RPC services.

There are a number of licenses and installation components you can select for the IBM DCE software:

- To run call processing applications on your Solaris system, you require the DCE base services license and you need to install the **client** component.
- If you also want to develop call processing applications on your Solaris system, you additionally require the DCE Application Development Tools license and you need to install the **appdev** component.

The Call Processing Software does not require you to configure these components.

Note that the network protocol TCP/IP is provided as part of Solaris. Check that the call processing server is configured to communicate with your Solaris system using TCP/IP (see Section 2.4).

10.3.1 Defining the Call Processing Server Name and Address

To ensure that the call processing server and your Solaris system can communicate, the name and network address of the call processing server must be defined either in your network server database (if your network has one) or on your Solaris system. For example, for a TCP/IP network, you can add the server's TCP/IP name and address to the hosts file.

For more information, refer to your network management documentation or see your network manager.

10.4 Installation Procedure

Follow these steps:

1. Log in as a superuser (login name `root`) on the local system.

For more information about superuser privileges, refer to your Solaris system manager documentation.

2. Create a directory for the call processing API software kit and change to that directory. For example:

```
# mkdir /ctc_pkg
# cd /ctc_pkg
```

3. Copy the file CTC600.SUN from the call processing server. For example, using FTP:

```
# ftp
ftp> open server_name
```

where *server_name* is the name of the server system. Enter *username* and *password* when prompted and then locate the directory used for the call processing server software installation. The Solaris kit is located in the subdirectory `\clients\solaris`.

For example:

```
ftp> cd C:\Program Files\Intel\Ctc\clients\solaris
```

Enter the following commands to copy the file:

```
ftp> binary
ftp> get ctc600.sun
ftp> quit
```

4. Unpack the kit from the tape archive file:

```
# tar -xvf ctc600.sun
```

5. Use the **pkgadd** utility to install the software. For example:

```
# pkgadd -d /ctc_pkg
```

For more information on **pkgadd**, refer to your Solaris system management documentation.

10.5 Files Installed

Table 10–1 describes the files installed during the call processing API installation procedure and their locations.

Table 10–1 Call Processing API Files Installed on Solaris

Description	Location
Call Processing Software definitions files to include in your application	/usr/include/
Executables for the Call Processing Software utilities	/usr/bin/
Call processing API and management API runtime libraries	/usr/lib/
Source code for the example application	/opt/intel/ctc600/ctc/example

10.6 What to Do Next

When the installation is complete:

1. If you did not read the release notes installed on the call processing server, read them now. See Section 2.8.1.
2. Check that your Solaris system can communicate with the call processing server and the switch. See Section 10.7.

10.7 Checking Communication

You use the CTC Test Program to check communication between your Solaris call processing client, the call processing server, and the switch.

Before you start CTC Test, make sure that the communications link between the call processing server and the switch is available and ready for use. For more information, refer to Section 3.10.

For detailed information about CTC Test, refer to the *Intel NetMerge Call Processing Software CTC Test User's Guide* (provided as a PDF file).

10.7.1 Procedure

Follow these steps:

1. Gather the following information:

Item	Description
DN	A telephone number on the switch that you can use to check the link. You assign a channel to this DN when you run CTC Test.
Server Name	The name of the call processing server system. For example, BOWFELL.

Item	Description
Logical Identifier	The logical identifier of the link between the server and the switch. For example, SERV1.
Network Transport	A value used to identify the protocol used between the call processing server and the call processing client. Specify the value: ncacn_ip_tcp

2. Enter the following command:

```
# ctc_test
```

The command prompt `ctcTest>` will appear.

3. Enter the **assign** command at the prompt, in the following format:

```
ctcTest> assign dn server_name logical_identifier ncacn_ip_tcp
```

Use the data you gathered in Step 1 for the command parameters. For example:

```
# ctcTest> assign 2345 skiddaw ctcserv1 ncacn_ip_tcp
```

This command assigns a channel to DN **2345**. The server name is `skiddaw` and the logical identifier for the link between the server and the switch is `ctcserv1`. The required value for the network transport is `ncacn_ip_tcp`.

10.7.2 Using CTC Test with Device Level Authorization

You need to use the **secassign** command instead of the **assign** command if both of the following are true:

- In the Configuration Program, you have enabled both Device Level Authorization, and either System or Application password checking. Refer to Chapter 3, Table 3-1.
- You have configured one or more authorized users, as described in Chapter 4.

Note If you have enabled Device Level Authorization but have selected None for password checking, you should use the `assign` command rather than the `secassign` command.

10.7.2.1 Procedure

Follow these steps:

1. Gather the information listed in Section 10.7.1, Step 1. In addition, you must supply a valid user name and password for an authorized user. Refer to Sections 4.2.1 and 4.2.2 for details.
2. Start CTC Test as described in Section 10.7.1, Step 2.
3. Enter the **secassign** command at the prompt:

```
ctcTest> secassign dn server_name logical_identifier ncacn_ip_tcp
```
4. You are prompted for a user name. Enter the name of an authorized user.
5. You are prompted for a password. Enter the user's password.

10.7.3 What to Do If CTC Test Fails

If CTC Test returns a fail message, check that:

- All details you specified are correct. The specified telephone number must be a valid number on the switch.
- The call processing server software is installed and running on the call processing server.
- The name for the call processing server is defined in your network name database or on your local call processing client system.
- If you are using **secassign**, check that the user name and password are configured in the authorization database, and are associated with the device, the link and the client that you specified.
- The link between the call processing server and the switch is enabled.
- The network connection between the call processing client and the call processing server is up and running.

If the procedure still fails, start the Control Program on the server and click the **Version** button to check the version of the server software.

The server must run Call Processing Software V6.0 software for compatibility with clients running the API V3.0, V4.0, V5.0 or V6.0 software. Refer to Section 1.3 for information about upgrading the server software.

If the procedure still fails, contact Intel.

10.8 Removing the Software

If you need to remove the call processing API software from your system, remove the existing software package:

```
# pkgrm CTC
```

10.9 Reinstalling the Call Processing API Software

To reinstall the call processing API software, remove the software (see Section 10.8) and follow the procedure in Section 10.4.

Installing the Call Processing API on Tru64 UNIX

11.1 Overview

The following table provides an overview of the contents of this chapter.

For details of...	See Section...
Hardware requirements	11.2
Software requirements	11.3
Installation instructions	11.4
Files installed	11.5
What to do after installation	11.6
Checking communication	11.7
Removing the software	11.8

Note that the management API is installed as part of the call processing API software kit. For more information, refer to Section 1.2.3.

11.2 Hardware Requirements

The following hardware is required:

- A Tru64 UNIX system in the same network as the call processing server. For more details of the hardware required, contact Compaq or your hardware supplier.
- 1 Mbyte of free hard disk space to load the software.

These are the minimum requirements for installing and using the call processing API and the call processing management API on a Tru64 UNIX system. To run additional software, you may need more resources, for example, additional disk space and memory.

11.3 Software Requirements

The following software is required:

- Tru64 UNIX Version 5.1.
- Basic Networking (TCP/IP) Services subset of Tru64 UNIX software.
- Tru64 UNIX DCE Runtime Services to support the RPC services.

The distributed call processing client/server uses the DCE runtime services for RPC to provide a common API across different platforms.

- If you are writing a call processing application on your Tru64 UNIX system, you also require the DCE Application Development Kit. This kit is a subset of the Tru64 UNIX Distributed Computing Environment for Tru64 UNIX Alpha Version 4.0 software.

11.3.1 Defining the Call Processing Server Name and Address

To ensure that the call processing server and your Tru64 UNIX system can communicate, the name and network address of the server must be defined either in your network server database (if your network has one) or on your call processing client. For example, for a TCP/IP network, you can add the call processing server's TCP/IP name and address to the hosts file.

For more information, refer to your network management documentation or see your network manager.

11.4 Installation Procedure

Follow these steps:

1. Log in as a superuser (login name `root`) on the local system.

For more information about superuser privileges, refer to your Tru64 UNIX system manager documentation.

2. Create a directory for the call processing API software kit and change to that directory. For example:

```
# mkdir /ctc_subset
# cd /ctc_subset
```

3. Copy the CTC600.TAR file from the call processing server. For example, using TCP/IP:

```
# ftp
ftp> open server_name
```

where *server_name* is the name of the call processing server.

Enter *username* and *password* when prompted and then locate the directory used for the call processing server software installation. The Tru64 UNIX kit is located in the subdirectory `\clients\600tru64`.

For example:

```
ftp> cd c:\Program Files\intel\ctc\clients\tru64
```

Enter the following commands to copy the file:

```
ftp> binary
ftp> get ctc600.tar
ftp> quit
```

4. Restore the call processing API kit from the CTC600.TAR file. Enter:

```
# tar -xf ctc600.tar
```

5. Enter a `setld` command that specifies the `-l` (load) function and the name of the directory that contains the software kit. For example, for `/CTC_SUBSET`:

```
# setld -l /ctc_subset
```

6. The `setld` utility prompts you to confirm that you want to install call processing API software. When you confirm this, it installs the subsets and verifies the installation.

Section 11.4.1 contains guidance if the Installation Verification Procedure (IVP) failed during the installation.

11.4.1 The IVP Fails

The installation procedures for the call processing API software on Tru64 UNIX contain an IVP which checks that the files have been correctly installed. If the IVP fails, take the following actions:

1. Check that the system quotas or resources on your system are adequate.
2. Check that the prerequisite software is installed.

3. Ensure that the hardware on your system meets requirements.
4. Check the contents of the following file:

```
# /usr/var/adm/smlogs/fverify.log
```

This file contains a log of the verification process and may help you diagnose the problem. For more information, refer to the diagnostics section of the Tru64 UNIX documentation.

5. Check that there are no problems with the system disk you are using.

If none of the above actions isolates the problem, contact Intel.

11.5 Files Installed

This section lists the files and symbolic links created during the installation procedure on a Tru64 UNIX system.

The installation procedure creates the directory tree shown in Table 11–1, in the system root directory (/) by default.

If you use the `setld -D` command to install the software, the `usr` subdirectory is copied to the root directory that you specify. For more information about the `setld -D` command, refer to your Tru64 UNIX documentation.

Table 11–1 Call Processing API Files Installed on Tru64 UNIX

Description	Location
Executables for the Call Processing Software utilities	/usr/opt/CTC600/bin/
Source code for the example application	/usr/opt/CTC600/example/
Call processing definitions files to include in your application	/usr/opt/CTC600/include/
Call processing API and call processing management API runtime libraries	/usr/opt/CTC600/shlib/

The installation procedure creates symbolic links to the files in the /usr/opt/CTC600 directory. Table 11–2 shows the default symbolic links.

Table 11–2 Tru64 UNIX Symbolic Links

/usr/bin/	ctc_cp ctc_test
/usr/include/	ctcdef.h ctc_code.h ctc_err.h ctc_mgmt.h ctc_rpc.h
/usr/shlib/	libctc_api.so libctcmgtapi.so

For more information about symbolic links, refer to your Tru64 UNIX documentation.

11.6 What to Do Next

When the installation is complete:

1. If you did not read the release notes installed on the call processing server, read them now. See Section 2.8.1.
2. Check that your Tru64 UNIX system can communicate with the call processing server and the switch. See Section 11.7.

11.7 Checking Communication

You use the CTC Test Program to check communication between your Tru64 UNIX call processing client, the call processing server, and the switch.

Before you start CTC Test, make sure that the communications link between the call processing server and the switch is available and ready for use. For more information, refer to Section 3.10.

For detailed information about CTC Test, refer to the *Intel NetMerge Call Processing Software CTC Test User's Guide* (provided as a PDF file).

11.7.1 Procedure

Follow these steps:

1. Gather the following information:

Item	Description
DN	A telephone number on the switch that you can use to check the link. You assign a channel to this DN when you run CTC Test.
Server Name	The name of the call processing server system. For example, BINSEY.
Logical Identifier	The logical identifier of the link between the server and the switch. For example, SERV1.
Network protocol	A value used to identify the protocol used between the call processing server and the call processing client. Specify the following value: ncacn_ip_tcp

2. Enter the following command:

```
# ctc_test
```

The command prompt `ctcTest>` will appear.

3. Enter the **assign** command at the prompt, in the following format:

```
ctcTest> assign dn server_name logical_identifier ncacn_ip_tcp
```

Use the data you gathered in Step 1 for the command parameters. For example:

```
# ctcTest> assign 2345 skiddaw ctcserv1 ncacn_ip_tcp
```

This command assigns a channel to DN **2345**. The server name is `skiddaw` and the logical identifier for the link between the server and the switch is `ctcserv1`. The required value for the network transport is `ncacn_ip_tcp`.

11.7.2 Using CTC Test with Device Level Authorization

You need to use the **secassign** command instead of the **assign** command if both of the following are true:

- In the Configuration Program, you have enabled both Device Level Authorization, and either System or Application password checking. Refer to Chapter 3, Table 3-1.

- You have configured one or more authorized users, as described in Chapter 4.

Note: If you have enabled Device Level Authorization but have selected None for password checking, you should use the `assign` command rather than the `secassign` command.

11.7.2.1 Procedure

Follow these steps:

1. Gather the information listed in Section 11.7.1, Step 1. In addition, you must supply a valid user name and password for an authorized user. Refer to Sections 4.2.1 and 4.2.2 for details.
2. Start CTC Test as described in Section 11.7.1, Step 2.
3. Enter the **secassign** command at the prompt:

```
ctcTest> secassign dn server_name logical_identifier ncaen_ip_tcp
```
4. You are prompted for a user name. Enter the name of an authorized user.
5. You are prompted for a password. Enter the user's password.

11.7.3 What to Do If CTC Test Fails

If CTC Test returns a fail message, check that:

- All details you specified are correct. The specified telephone number must be a valid number on the switch.
- The call processing server software is installed and running on the call processing server.
- The name for the call processing server is defined in your network name database or on your local call processing client system.
- If you are using the `secassign` command, check that the user name and password are configured in the authorization database, and are associated with the device, the link and the client that you specified.
- The link between the call processing server and the switch is enabled.
- The network connection between the call processing client and the call processing server is up and running.

If the procedure still fails, start the Control Program on the server and click the **Version** button to check the version of the server software.

The server must run Call Processing Software V6.0 software for compatibility with clients running the API V3.0, V4.0, V5.0 or V6.0 software. Refer to Section 1.3 for information about upgrading the server software.

If the procedure still fails, contact Intel.

11.8 Removing the Software

The following command deletes the call processing API software and all symbolic links and subset files:

```
# setld -d CTCAPI600
```

Installing the Call Processing API on Windows Systems

12.1 Overview

This chapter describes how to install the call processing API on Windows 9x, Windows NT, Windows 2000 and Windows XP systems.

The following table provides an overview of the contents of this chapter.

For details of...	See Section...
Hardware requirements	12.2
Software requirements	12.3 and 12.4
Installation instructions	12.5
Files installed	12.6
What to do after installation	12.7
Checking communication	12.8
Removing the software	12.9
Reinstalling the software	12.10

Note that the call processing management API is installed as part of the call processing API software kit. For more information, refer to Section 1.2.3.

12.2 Hardware Requirements

The following hardware is required:

- A PC in the same network as the call processing server, suitable for installing your Windows operating system. For more details of the hardware required, contact Microsoft or your hardware supplier.
- At least 2 Mbytes of free disk space for installing and using the call processing API.

- A network adapter card for connection to the call processing server. For suitable cards, refer to the documentation provided with your system, see your hardware supplier or refer to the Microsoft web site (for example, to the *Windows Hardware Compatibility List*).

These are minimum requirements for installing and using the call processing API and the management API on a Windows system. If you need to run additional software on your Windows system, remember to check the requirements for that software. You may need additional resources, for example, additional memory.

12.3 Software Requirements

This section lists the software required for each operating system.

12.3.1 Windows 9x

Windows 95, Windows 98 or Windows Me must be running on your PC.

12.3.2 Windows NT

One of the following must be running on your PC:

- Windows NT Workstation Version 4.0 (SP6 or later)
- Windows NT Server Version 4.0 (SP6 or later)

12.3.3 Windows 2000

Windows 2000 must be running on your PC.

12.3.4 Windows XP

Windows XP must be running on your PC.

12.3.5 All Windows Systems

You must install one of the following network protocols, for communication with your call processing server:

- Named pipes
- NetBIOS† over NetBEUI
- NetBIOS over TCP/IP
- Novell†
- SPX
- TCP/IP

You can install these network protocols as part of your Windows installation. Make sure you install the network protocol you require for communication with the call processing server. For more information about installing these network protocols, refer to your Windows installation documentation.

12.4 Other Installation Requirements

This section lists other requirements that you need to know about when installing the call processing API.

12.4.1 Defining the Call Processing Server Name and Address

To ensure that your Windows system can communicate with the call processing server, you must define the call processing server's name and network address on one of the following:

- Your network/name server database (if your network has one)
- Your call processing client

For example, for a TCP/IP network, you can add the call processing server's TCP/IP name and address to the hosts file.

For more information, refer to your network management documentation or see your network manager.

12.5 Installation Procedure

Follow these steps:

1. Log in as administrator. The installation procedure requires administrator privileges.
2. Create a temporary working directory to receive the installation kit. For example, C:\Cttemp.
3. Connect to the call processing server installation drive (using Windows Explorer) and copy the following file to your temporary working directory:

```
drive:\directory\Clients\W32\CTCI86V6.EXE
```

where *drive:\directory* is the drive and directory used for the call processing server software installation. For example, C:\Program Files\Intel\Ctc.

4. Run CTCI86V6.EXE from your temporary working directory. This installs the call processing API software onto your Windows system. See Section 12.6 for details of the files installed by this procedure.

5. Reboot your PC.

12.6 Files Installed

Table 12–1 lists the files installed during the installation procedure.

Table 12–1 Call Processing API Files Installed on Windows Systems

Description	Location
Source code for the example application	<i>drive:\directory\EXAMPLES\</i>
Call processing API and management API libraries	<i>drive:\directory\LIB\</i>
Call Processing Software executables and help files	<i>drive:\directory\BIN\</i>
Call Processing Software definitions files to include in your application	<i>drive:\directory\INCLUDE\</i>

where *drive:\directory* is the drive and directory used for the call processing API software installation.

12.7 What to Do Next

When the installation is complete:

1. If you did not read the release notes installed on the call processing server, read them now. See Section 2.8.1.
2. Check that your Windows system can communicate with the call processing server and the switch. See Section 12.8.

12.8 Checking Communication

You use the CTC Test program to check communication between your Windows call processing client, the call processing server, and the switch.

Before you start CTC Test, make sure that the communications link between the call processing server and the switch is available and ready for use. For more information, refer to Section 3.10.

For detailed information about CTC Test, refer to the *Intel NetMerge Call Processing Software CTC Test User's Guide* (provided as a PDF file).

12.8.1 Procedure

Follow these steps:

1. Gather the following information:

Item	Description
DN	A telephone number on the switch that you can use to check the link. You assign a channel to this DN when you run CTC Test.
Server name	The name of the call processing server system. For example, SKIDDAW.
Logical identifier	The logical identifier of the link between the server and the switch. For example, SERV1.
Network protocol	A value used to identify the protocol used between the call processing server and the call processing client. Specify one of the following values (in lower case):
For this protocol...	Specify the value...
TCP/IP	ncacn_ip_tcp
NetBIOS over NetBEUI	ncacn_nb_nb
NetBIOS over TCP/IP	ncacn_nb_tcp
Named Pipes	ncacn_np
Novell SPX	ncacn_spx
TCP universal datagram protocol	ncadg_ip_udp
Local RPC	ncalrpc ¹
¹ If the call processing client is installed on the same PC as the call processing server. Note that this is not supported on Windows 9x.	

2. Start CTC Test. From the Start menu, select the following:

Programs → Intel NetMerge Call Processing API → ctcTest

The command prompt `ctcTest>` will appear.

3. Enter the **assign** command at the prompt, in the following format:

```
ctcTest> assign dn server_name logical_identifier network_protocol
```

Use the data you gathered in Step 1 for the command parameters. For example:

```
# ctcTest> assign 2345 skiddaw ctcserv1 ncacn_spx
```

This command assigns a channel to DN **2345**. The server name is `skiddaw` and the logical identifier for the link between the server and the switch is `ctcserver1`. The network protocol is `ncacn_spx`.

12.8.2 Using CTC Test with Device Level Authorization

You need to use the **secassign** command instead of the **assign** command if both of the following are true:

- In the Configuration Program, you have enabled both Device Level Authorization, and either System or Application password checking. Refer to Chapter 3, Table 3-1.
- You have configured one or more authorized users, as described in Chapter 4.

Note: If you have enabled Device Level Authorization but have selected None for password checking, you should use the `assign` command rather than the `secassign` command.

12.8.2.1 Procedure

Follow these steps:

1. Gather the information listed in Section 12.8.1, Step 1. In addition, you must supply a valid user name and password for an authorized user. Refer to Sections 4.2.1 and 4.2.2 for details.
2. Start CTC Test as described in Section 12.8.1, Step 2.
3. Enter the `secassign` command at the prompt:

```
ctcTest> secassign dn server_name logical_id network_protocol
```
4. You are prompted for a user name. Enter the name of an authorized user.
5. You are prompted for a password. Enter the user's password.

12.8.3 What to Do If CTC Test Fails

If CTC Test returns a fail message, check that:

- All details you specified are correct. The specified telephone number must be a valid number on the switch.
- The call processing server software is installed and running on the server.
- The name for the call processing server is defined in your network name database or on your local call processing client system.

- If you are using the secassign command, check that the user name and password are configured in the authorization database, and are associated with the device, the link and the client that you specified.
- The link between the call processing server and the switch is enabled.
- The network connection between the call processing client and the call processing server is up and running.

If the procedure still fails, start the Control Program on the server and click the **Version** button to check the version of the server software.

The call processing server must run Call Processing Software V6.0 server software for compatibility with clients running the API V3.0, V4.0, V5.0 or V6.0 software. Refer to Section 1.4 for information about upgrading the call processing server software.

If the procedure still fails, contact Intel.

12.9 Removing the Software

This section describes how to remove the API software. Follow the appropriate procedure for your operating system.

If, after removing the software, you want to reinstall, reboot your system before starting the installation procedure.

12.9.1 Windows 9x and Windows NT

1. From the Control Panel, open the Add/Remove Programs dialog box.
2. Select the **Install/Uninstall** tab.
3. Select **Intel NetMerge Call Processing API** from the list of software.
4. Click on the **Add/Remove** button.

12.9.2 Windows 2000

1. From the Control Panel, open the Add/Remove Programs dialog box.
2. Click the Change or Remove Programs icon.
3. Select **Intel NetMerge Call Processing API** from the list of software.
4. Click on the **Remove** button.

12.9.3 Windows XP

1. From the Control Panel, select Add or Remove Programs.

2. Select **Intel NetMerge Call Processing API** from the list of software.
3. Click on the **Remove** button.

12.10 Reinstalling the Call Processing API

If you reinstall the call processing API V6.0 software, make sure that you exit all Call Processing Software programs. If a Call Processing Software program is running in a separate window, exit the window.

If you do not exit all Call Processing Software programs before you try to reinstall the product, the installation fails.

Installing the Call Processing API for Java

13.1 Overview

The following table provides an overview of the contents of this chapter.

For details of...	See Section...
Hardware requirements	13.2
Software requirements	13.3
Installation procedure	13.4
Files installed	13.5
What to do next	13.6
Checking communication	13.7
Removing the software	13.8

Note that the management API is installed as part of the call processing API software kit. For more information, refer to Section 1.2.3.

13.2 Hardware Requirements

To install the call processing API, you require:

- A system suitable for running the Java 2 Runtime Environment (JRE). For more details of the hardware required, contact your hardware supplier.

The system needs to have network access to the call processing server. For example, across a local area network.

- 1 Mbyte of free hard disk space to install the call processing API.

These are minimum requirements for installing and using the call processing API and the call processing management API. To run additional software, you may need more resources, for example, additional disk space and memory.

13.3 Software Requirements

Your system requires the following:

- To install the call processing API, Java 2 compatible Runtime Environment (JRE), Standard Edition, V1.2 or later.
- If you want to develop call processing Java applications, the Java Software Development Kit (SDK), Standard Edition, V1.2 or later.

Note: If you wish to use the V6.0 user name and password authorization feature in your Java client application, you are required to install the Java Cryptography Extensions (JCE). The JCE is integrated into the Java 2 SDK, V1.4. Alternatively, you can use an earlier SDK version and download the JCE separately. For more information about the JCE, see the web page <http://java.sun.com/products/jce/index.html>.

Refer to Chapter 4 for more information about the authorization feature.

- The TCP/IP network protocol you require for communication with the call processing server. Section 2.4 describes the software requirements for the call processing server.

It is recommended that you use the version of the JRE or Java SDK shipped by Sun as the Java 2 Platform. It is possible to install a runtime and development environment from a manufacturer other than Sun, but the environment must implement the Java 2 specification, including support for RMI. The descriptions in the remainder of this chapter assume that you are using the Java 2 Platform, or its equivalent.

13.3.1 Defining the Call Processing Server Name and Address

To ensure that the server and your client system can communicate, the name and network address of the server must be defined either in your network server database (if your network has one) or on your client system.

For more information, refer to your network management documentation or see your network manager.

13.4 Installation Procedure

This section describes how to make the call processing API JAR files available to client systems so that these systems can communicate with the server and run a call processing Java application. There are two methods.

Method 1: Copy the JAR files from the call processing server and include them in the JVM classpath on the client. You must follow this

method if you intend to develop your call processing Java application on the client.

Method 2: You can only use this method if you have already developed your call processing application, and it is in the form of a Java applet. You install the JAR files and your call processing applet on the web server so that they become available to the client from a web browser on the client.

Section 13.4.1 lists the files you need to copy. Section 13.4.2 describes the installation procedure for Method 1. Section 13.4.3 describes the installation procedure for Method 2.

13.4.1 Required Files

You need to copy files as follows:

- For Method 1, copy the files in Table 13–1 from the server to your client JVM.
- For Method 2, copy the files in Table 13–1 to the Web server directory that also contains your call processing applet class files.
- If the client is an external client that will use the call processing API to communicate across the Internet with your call processing server, you also need to copy the files in Table 13–2. Refer to Chapter 6 for more information about external clients.
- If you are going to develop call processing applications for the Java platform, you should also copy the programming reference information in Table 13–3. The files listed contain the documentation for both the call processing API and the call processing management API.

In all the tables, *drive\directory* is the drive and directory used for the server software installation. The default installation directory on the server is C:\Program Files\Intel\Ctc.

Table 13–1 Java Client API Files and Supporting Files

File Name	File Location on Call Processing Server
CTCAPI.JAR	<i>drive:\directory\clients\java</i>
CTCMGMTAPI.JAR	<i>drive:\directory\clients\java</i>
CTCSAMPLE.JAR	<i>drive:\directory\clients\java</i>
CTCSAMPLE.ZIP	<i>drive:\directory\clients\java</i>
CTCTEST.JAR	<i>drive:\directory\clients\java</i>

Table 13–2 External Access Client Files

File Name	Location
SOAP.JAR	<i>drive:\directory\java</i>
XERCESSIMPL.JAR	<i>drive:\directory\java</i>
XMLPARSERAPIS.JAR	<i>drive:\directory\java</i>
MAIL.JAR	<i>drive:\directory\java</i>
ACTIVATION.JAR	<i>drive:\directory\java</i>

Table 13–3 Documentation Files

File Name	Location
One of the files listed below. The extension depends on your client's operating system.	<i>drive:\directory\clients\java</i>
<ul style="list-style-type: none">• CTCAPIDOC.ZIP• CTCAPIDOC.TARZ	

13.4.2 Method 1

This section describes how to install the call processing API and management API on the client by including the call processing JAR files in the Java Virtual Machine (JVM) classpath on the client.

Follow these steps:

1. Log into your system as a user with sufficient privileges. For example, on a Windows NT system, log in as Administrator. For more information about privileges, refer to your system management documentation.
2. Include the call processing JAR files in the JVM classpath on your client by copying them into the installed extensions directory of the JRE. The files are described in Section 13.4.1.

The default extensions directory varies according to the Java version. For example, on a Windows NT client, with the JRE from Sun's Java 2 Platform V1.3 installed, the default extensions directory is:

```
C:\Program Files\Javasoft\JRE\1.3\LIB\EXT
```

3. If you are also going to develop Java applications:

- So that you can compile your Java application, put additional copies of the files in the JDK directory.

The JDK directory varies according to the Java version. For example, on a Windows NT call processing client, with the JDK from Sun's Java 2 Platform V1.3 installed, the default directory is:

```
C:\JDK1.3\JRE\LIB\EXT
```

- Unpack or unzip the Java reference information for the Call Processing Software. The files are listed in Table 13–3.

You read the Java reference information with your web browser. To access the information, open the file INDEX.HTML from within your browser or by double-clicking on it.

13.4.3 Method 2

This section describes how to make the call processing JAR files and a call processing applet available to a web browser on the client.

13.4.3.1 Before You Begin

The installation procedure for Method 2 assumes the following:

- The call processing server is installed on the same system as the Web server.
- You have already developed your call processing application as an applet, and an HTML page to run the applet.

Note that to develop your call processing applet, you must first install the call processing API by following the Method 1 installation procedure described in Section 13.4.2.

- The call processing applet is in a Web server directory on the call processing server. This is because Java applets can only communicate with the machine from which they were loaded.

Note: files containing the source code for a sample Java application can be found in the following location:

```
drive:\directory\CLIENTS\JAVA\CTCSAMPLE.ZIP
```

where *drive:\directory* is the installation drive and directory. By default, this is:

```
C:\PROGRAM FILES\INTEL\CTC
```

The CTCSAMPLE.ZIP file also contains the file SAMPLEREADME.TXT, which describes how to create the application.

There is also a compiled example call processing applet in the `CTCSAMPLE.JAR` file in the same location. This file implements an example downloadable applet and the required HTML.

Securing the JAR Files and the Applet

Note that Method 2 could make the call processing JAR files and the call processing applet available to any system that can access the HTML pages on the call processing server (for example, across the Internet). Make sure that you restrict access to these files so that only authorized clients can access them; for example, by configuring the Web server to restrict external access.

13.4.3.2 Procedure

Follow these steps:

1. On the call processing server, copy the files listed in Table 13–1 and (if desired) Table 13–2 to the Web server directory that also contains your call processing applet class files.
2. In the HTML page for the call processing applet, specify the following for use on the call processing client:
 - The class file
 - The relative directory structure
 - The call processing JAR files

If your call processing applet requires Java 2 Platform functionality, the client needs to provide a Java 2 Platform JVM either in the web browser or external to the web browser.

To provide the Java 2 Platform JVM externally, install the Java 2 Runtime Environment (JRE) plug-in on the client. You can download this plug-in from the Sun website. For example, you can download the JRE 2 V1.4 from this address:

<http://java.sun.com/products/plugin/index-1.4.html>

Alternatively, if you open the HTML file for the example applet provided in the Call Processing Software kit, this will automatically download the plug-in.

13.5 Files Installed

All of the call processing API and management API files for Java are installed in

one location on the client:

drive:\directory

where *drive:\directory* is the location of the installed extensions directory of your client's JRE. For example, on a Windows NT system, with the JRE from Sun's Java 2 Platform V1.3 installed, the default location is:

C:\Program Files\javasoft\JRE\1.3\LIB\EXT

13.6 What to Do Next

When the installation is complete:

1. If you did not read the release notes installed on the call processing server, read them now. See Section 2.8.1.
2. Check that your call processing client system can communicate with the call processing server and the switch. See Section 13.7.

13.7 Checking Communication

You use the CTC Test program to check communication between your call processing client, the call processing server, and the switch.

Before you start CTC Test, make sure that the communications link between the call processing server and the switch is available and ready for use. For more information, refer to Section 3.10.

For detailed information about CTC Test, refer to the *Intel NetMerge Call Processing Software CTC Test User's Guide* (provided as a PDF file).

13.7.1 Procedure

This section describes how to run CTC Test in the JRE from the Java 2 Platform. Follow these steps:

1. Gather the following information:

Item	Description
DN	A telephone number on the switch that you can use to check the link. You assign a channel to this DN when you run CTC Test.
Server Name	The name of the call processing server system. For example, GABLE.
Logical Id	The logical identifier of the link between the server and the switch. For example, SERV1.

2. Start CTC Test by entering the following command:

```
java -jar ctctest.jar
```

The command prompt `ctcJavaTest>` will appear.

3. Enter the **assign** command at the prompt in the following form:

```
ctcJavaTest> assign dn server_name logical_id
```

Use the data you gathered in Step 1 for the command parameters. For example:

```
ctcJavaTest> assign 2345 skiddaw ctcserv1
```

This command assigns a channel to DN **2345**. The server name is **skiddaw**. The logical identifier for the link between the server and the switch is **serv1**.

13.7.2 Using CTC Test with Device Level Authorization

You will need to enter a user name and password with the **assign** command if both of the following are true:

- In the Configuration Program, you have enabled both Device Level Authorization, and either System or Application password checking. Refer to Chapter 3, Table 3-1.
- You have configured one or more authorized users, as described in Chapter 4.

Note: If you have enabled Device Level Authorization but have selected None for password checking, do not enter a user name and password when using the **assign** command.

13.7.2.1 Procedure

Follow these steps:

1. Gather the information listed in Section 13.7.1, Step 1. In addition, you must supply a valid user name and password for an authorized user. Refer to Sections 4.2.1 and 4.2.2 for details.
2. Start CTC Test as described in Section 13.7.1, Step 2.
3. Enter the **assign** command at the prompt in this form:

```
ctcJavaTest> assign dn server_name logical_id username password
```

13.7.3 Using CTC Test On External Java Clients

You will need to enter a Web server URL and port number in the **assign** command if the following are true:

- The Java client will communicate with the call processing server across the Internet through a firewall, using SOAP (the Simple Object Access Protocol).
- You have carried out any necessary additional configuration, as described in Chapter 6.
- You have copied the files listed in Table 13–2 to your Java client, as described in Section 13.4.2 or Section 13.4.3.

Refer to Chapter 6 for more information about SOAP and external Java clients.

13.7.3.1 Procedure

Follow these steps:

1. Gather the information listed in Section 13.7.1, Step 1. In addition, you must supply:
 - The URL of the Web server used to access the call processing server.
 - The port number used to reach the Web server. The default is 80.
2. Start CTC Test as described in Section 13.7.1, Step 2.
3. Enter the **assign** command at the prompt in this form:

```
ctcJavaTest> assign dn server_name logical_id url port-number
```

where *url* is the URL of the Web server and *port-number* is the port number.

13.7.4 What to Do If CTC Test Fails

If CTC Test returns a fail message, check that:

- All details you specified are correct. The specified telephone number must be a valid number on the switch.
- The call processing server software is installed and running on the call processing server.
- The name for the call processing server is defined in your network name database or on your local call processing client system.
- If you entered a user name and password, check that they are configured in the authorization database, and are associated with the device, the link and the client that you specified.

- The link between the call processing server and the switch is enabled.
- The network connection between the call processing client and the call processing server is up and running.

If the procedure still fails, start the Control Program on the server and click the **Version** button to check the version of the server software.

The server must run Call Processing Software V6.0 software for compatibility with clients running the API V3.0, V4.0, V5.0 or V6.0 software. Refer to Section 1.3 for information about upgrading the server software.

If the procedure still fails, contact Intel.

13.8 Removing the Software

If you need to remove the call processing API and management API software from your system, simply delete the call processing JAR files from the locations to which you copied them during the Method 1 or Method 2 installation procedure.

What to Do Next

When the installation is complete, your call processing clients are ready to run call processing applications.

You can now do one of the following:

1. If you have not yet developed your call processing application, refer to the programming information in the following documentation:
 - *Intel NetMerge Call Processing Software C Programming Guide*
 - *Intel NetMerge Call Processing API for the Java Platform* (provided as HTML files)
2. If you have already developed your call processing application, install it on your call processing client systems.

Index

A

Access number length, 5-6
ACSE, 3-9
Advanced settings, 3-15
Agent device type, 4-6
Application
 creating, 1-2
 management, 1-4
 password, 3-6, 4-3, 4-5
 user name, 4-5
Application data, 3-8, 5-1
Application identifier, 3-13, 3-14
Association Control Service Element (ACSE), 3-9
auth_imp_exp.exe, 4-10
Authorization database, 3-6, 4-1
 checking user information, 4-2
Authorization file
 definition, 4-9
 example, 4-13
 format, 4-11
Authorization Setup Program, 3-2, 4-1 to 4-8
 starting, 4-7
Auto restart monitors, 3-5
Auto start link, 3-4, 7-4
Auto start trace
 link tracing, 3-6
 management tracing, 3-8
Avaya DEFINITY
 ISDN configuration information, 3-12
 TCP/IP configuration information, 3-12

B

Baud rate, 3-11

C

Call device type, 4-6
Call processing API
 installing on HP-UX, 8-1
 installing on OpenVMS, 9-1
 installing on Solaris, 10-1
 installing on the JRE, 13-1
 installing on Tru64 UNIX, 11-1
 installing on Windows, 12-1 to 12-8
 reinstalling on Solaris, 10-7
 reinstalling on Windows, 12-8
Call processing client
 See Client
Call processing server
 See Server
Check interval, 3-7
Client
 HP-UX, 8-1 to 8-6
 Java, 13-1 to 13-10
 OpenVMS, 9-1 to 9-6
 Solaris, 10-1 to 10-7
 Tru64 UNIX, 11-1 to 11-8
 Windows, 12-1 to 12-8
Client application
 programming interfaces, 1-2
 requirements for distributed data, 5-3
Client link number, 3-12
Client system name
 device level authorization, 4-7
Communications link
 See link

- Compact installation option, 2-9
- Configuration Program
 - leaving, 3-18
 - registry information, 3-2
 - starting, 3-2
 - to configure link, 3-1
 - when to use, 3-2
- Control Program, 1-4, 3-17, 7-4
 - changing link configuration, 3-17
 - starting, 3-17
 - starting a link manually, 3-4, 7-4
- CSTA
 - device query, 3-9
 - ISDN configuration information, 3-11
 - messages required for link state checking, 3-15
 - protocol, 2-2
 - TCP/IP configuration information, 3-10
 - V.24 configuration information, 3-11
 - X.25 configuration information, 3-11
- CTC Test
 - HP-UX clients, 8-4
 - Device Level Authorization, 8-5
 - Java clients, 13-7
 - Device Level Authorization, 13-8
 - OpenVMS clients, 9-6
 - Device Level Authorization, 9-7
 - Solaris clients, 10-4
 - Device Level Authorization, 10-5
 - Tru64 UNIX clients, 11-5
 - Device Level Authorization, 11-6
 - Windows clients, 12-4
 - Device Level Authorization, 12-6
- CTC trace level, 3-6
- ctcChannel object, 6-3
- CTCFULL trace level, 3-6
- ctcServerStarted event, 7-4
- Custom installation option, 2-9
- Customer number, 3-13, 3-14

D

- Data parameters on call processing server, 3-7
- DCE/RPC
 - OpenVMS clients, 9-2
 - Solaris clients, 10-2
 - Tru64 UNIX clients, 11-2
- DCE/RPC interface, 1-3
 - distributed data transport, 5-5
- DEFINITY
 - configuration information required, 3-12
 - supported network cards, 2-2
- Deleting symbolic links, 11-8
- Device, 3-11, 3-12, 3-14
 - V.24 connection, 3-11
 - X.25 connection, 3-11
- Device event buffers, 3-7
- Device Level Authorization, 3-2, 3-6, 4-1 to 4-14
 - CTC Test on HP-UX clients, 8-5
 - CTC Test on OpenVMS clients, 9-7
 - CTC Test on Solaris clients, 10-5
 - CTC Test on Tru64 UNIX clients, 11-6
 - CTC Test on Windows clients, 12-6
- Device number length, 5-5
- Device numbers, 4-6
 - multiple, 4-6
 - range, 4-7
- Device query, 3-9
- Device types
 - agent, 4-6
 - call, 4-6
 - DN, 4-6
 - physical id, 4-6
 - route, 4-6
- Dialup number for ISDN, 3-11
- Disk space
 - HP-UX, 8-1
 - Java, 13-1
 - OpenVMS, 9-1
 - Solaris, 10-1
 - Tru64 UNIX, 11-1
 - Windows client, 12-1
- DIST trace level, 3-6
- Distributed Computing Environment
 - See* DCE
- Distributed data configuration example, 5-6
- Distributed Data Setup Program, 3-2, 5-1 to 5-9
 - starting, 5-6
- Distribution
 - kit, 2-1

- media, 2-1
- DLA
 - See* Device Level Authorization
- DN device type, 4-6
- DTE
 - local DTE address, 3-14
 - remote DTE address, 3-14

E

- EiconCard
 - Connections for Windows 2000/XP, 2-4
 - Connections for Windows NT, 2-4
 - WAN Services for Windows NT, 2-4
- Errors
 - import/export of authorization file, 4-10
 - link startup, 7-4
- Event buffering, 3-7, 3-16
- Exporting an authorization file, 4-9

F

- File locations
 - HP-UX, 8-3
 - Java client, 13-6
 - OpenVMS, 9-5
 - Solaris, 10-3
 - Tru64 UNIX client, 11-4
 - Windows client, 12-4
- Firewall, passing SOAP messages through, 1-4, 6-1, 6-2
- Full access
 - device level authorization, 4-7
 - remote management, 3-8

H

- Host name, 3-13, 3-14
 - remote management, 3-8
- HP-UX client
 - 700-series system, 8-1
 - 800-series system, 8-1
 - disk space, 8-1
 - HP-UX version, 8-2
 - installation, 8-1 to 8-6
 - software required, 8-2

- HTTP
 - distributed data transport, 5-5
- HTTP transport, 6-1

I

- Import/export of authorization file, 4-9 to 4-14
- Installation
 - call processing API on HP-UX, 8-1 to 8-6
 - procedure, 8-2
 - call processing API on Java, 13-1 to 13-7
 - call processing API on OpenVMS, 9-1 to 9-6
 - call processing API on Solaris, 10-1 to 10-7
 - procedure, 10-2
 - call processing API on Tru64 UNIX, 11-1 to 11-5
 - procedure, 11-2
 - call processing API on Windows, 12-1 to 12-8
 - procedure, 12-3
 - call processing server and call processing API on same system, 1-3
 - call processing server software, 2-1 to 2-12
 - procedure, 2-8
 - compact, 2-9
 - custom, 2-9
 - privileges, 2-8
 - summary, 1-1
 - symbolic links on Tru64 UNIX, 11-4
 - typical, 2-9
- IP address
 - local, 3-10, 3-12, 3-13
 - secondary, 3-10
 - switch, 3-10, 3-12, 3-13
- ISDN, 3-11, 3-12

J

- Jakarta Tomcat
 - See* Tomcat
- Java API
 - installation, 13-1 to 13-7

- software required, 13-2
- Java client
 - disk space, 13-1
 - installation, 13-1 to 13-6
 - Java version, 13-2
 - software required, 13-2
 - using SOAP, 6-2
- Java RMI interface, 1-3
 - distributed data transport, 5-5
- Java servlet, 6-1

L

- LCN, 3-11
- Link
 - advanced settings, 3-15
 - check interval, 3-7
 - checking, 3-19
 - device buffers, 3-7
 - event buffering, 3-7, 3-16
 - monitor channel buffers, 3-7
 - route point buffering, 3-7
 - starting automatically, 3-4, 7-4
 - starting manually, 3-4, 7-4
 - startup problems, 7-4
 - state checking retry, 3-7
- Link name, 3-1, 3-4, 4-7
- Link state checking, 3-15
- Local
 - DTE address, 3-14
 - IP address, 3-10, 3-12, 3-13
 - link, 5-4
- Local link, distributed data, 5-2
- Local server, distributed data, 5-2
- Local user name, 4-4
- Log file
 - import/export authorization file errors, 4-10
 - SOAP format messages, 6-6 to 6-7
 - Tru64 UNIX installation, 11-4
- Logical Channel Number
 - See* LCN
- Logical identifier, 3-4, 3-17
 - definition, 3-1
 - in authorization file, 4-11
 - in trace file name, 3-7

- select in Authorization Setup Program, 4-7
- select in Control Program, 3-18

Login user name, 4-4

M

- Machine identifier, 3-13, 3-14
- Mail name, 3-14, 3-15
- Management API, 1-4
- Maximum files
 - link tracing, 3-7
 - management tracing, 3-8
- Maximum files action
 - link tracing, 3-7
 - management tracing, 3-8
- Maximum monitors, 3-4
- Meridian Link
 - software versions, 2-3
- MLP, 3-14
- Monitor access
 - device level authorization, 4-7
 - remote management, 3-8
- Monitor channels
 - event buffers, 3-7
- Monitors
 - auto restart, 3-5
 - maximum, 3-4

N

- NET trace level, 3-6
- Network adapter card
 - requirements, 2-2
 - software required, 2-3
- Network ID of switch, 5-5
- Network protocols
 - HP-UX clients, 8-2
 - Java API, 13-2
 - Java client, 13-2
 - Solaris clients, 10-2
 - Tru64 UNIX clients, 11-2
 - Windows client, 12-2
- Nortel Meridian 1, 2-3
 - TCP/IP configuration information, 3-13
 - X.25 configuration information, 3-14

O

- OpenVMS client
 - call processing API installation, 9-1 to 9-6
 - classes required, 9-2
 - communication with call processing server, 9-3
 - CTC_STARTUP.COM, 9-6
 - OpenVMS version, 9-2
 - running call processing API at startup, 9-6
 - startup file, 9-6
 - TCP/IP hosts file, 9-3

P

- Password
 - Device Level Authorization, 4-5
- Password checking, 3-6
 - application, 3-6
 - none, 3-6
 - system, 3-6
- Permission level, 4-7
- Physical Id device type, 4-6
- Port number
 - CSTA switch, 3-10
 - in CTC Test assign command, 6-7, 13-9
 - Meridian switch, 3-13
- Prefix
 - area code of switch, 5-5
- Privileges
 - HP-UX, 8-2
 - Java API, 13-4
 - OpenVMS, 9-2
 - Solaris, 10-2
 - Tru64 UNIX, 11-2
 - Windows, 12-3
- Problems
 - communication problems with Meridian 1, 3-14
 - link startup, 7-4
 - link state not ON, 3-19
 - OpenVMS installation, 9-5
 - Tru64 UNIX installation, 11-3

R

- Range
 - device numbers for authorization, 4-7
 - Registry
 - link information, 3-2
 - Release notes
 - location of README.TXT, 2-11
 - Remote DTE address, 3-14
 - Remote link, distributed data, 5-2, 5-4
 - Remote management
 - host name, 3-8
 - security, 3-8
 - security level, 3-8
 - user name, 3-8
 - Remote Procedure Calls
 - See* RPC
 - Remote server, distributed data, 5-2, 5-4
 - Retry count, 3-7
 - Route point
 - buffering, 3-7
 - device type, 4-6
 - RPC
 - OpenVMS clients, 9-2
-
- ## S
- Secondary IP address, 3-10
 - Security level, remote management, 3-8
 - Server
 - data parameters, 3-7
 - hardware requirements, 2-2
 - installation kit, 2-1
 - installing call processing server software, 2-1 to 2-12
 - reinstalling call processing server software, 2-13
 - software requirements, 2-3
 - swap call reference, 3-7
 - Server application data, 3-8
 - Server software
 - starting, 7-1
 - Servlet, 6-1
 - container, 6-1
 - rpcrouter, 6-1

setld
 -d command, 11-8
 load function, 11-3
 utility, 11-3
 Simple Object Access Protocol
 See SOAP
 SOAP
 configuration supported, 1-4
 configuring logging, 6-6 to 6-7
 description, 1-1, 6-1
 use by distributed data servers, 6-8
 use by Java client, 6-2, 6-7
 used by Call Processing Software, 6-1 to 6-8
 Solaris client
 call processing API installation, 10-1 to 10-7
 disk space, 10-1
 software required, 10-2
 Solaris version, 10-2
 Swap call reference, 3-7
 Switch
 Avaya DEFINITY, 3-12
 CSTA, 2-2
 CSTA Phase I, 3-9
 CSTA Phase II, 3-9
 CSTA Phase III, 3-9
 DEFINITY, 3-12
 IP address, 3-10, 3-12, 3-13
 ISDN connection, 3-11
 logical identifier, 3-4
 MLP protocol, 3-14
 Nortel Meridian 1, 2-3, 3-14
 Nortel Meridian SL-1, 2-3
 port number, 3-13
 V.24 connection, 3-11
 X.25 connection, 3-11, 3-14
 Switch option, 3-14
 Switch type
 CSTA, 5-6
 DEFINITY, 5-6
 Meridian, 5-6
 Symbolic links
 deleting, 11-8

SYSSUPDATE, 9-4
 System password, 3-6, 4-4
 System startup file on OpenVMS, 9-6

T

TCP/IP
 hardware required, 2-2
 Meridian Link version, 2-2
 required for distributed data networks, 5-3
 software required, 2-3
 Time stamp, 3-5
 Tomcat
 definition, 6-1
 integrating with Web servers, 6-5
 started as Windows service, 6-4
 Trace file name
 link tracing, 3-7
 management tracing, 3-8
 Trace file path
 link tracing, 3-7
 management tracing, 3-8
 Trace levels, 3-6
 Tracing
 auto start management tracing, 3-8
 link, 3-6
 management, 3-8
 Transport interfaces
 for distributed data servers, 5-5
 Tru64 UNIX client
 call processing API installation, 11-1 to 11-5
 hardware required, 11-1
 installation verification log, 11-4
 software required, 11-2
 Typical
 installation option, 2-9

U

Upgrading, 1-4
 URL, 6-3, 6-7
 URL of web server, 5-5
 User name
 and password checking, 4-3

- application, 4-5
- Device Level Authorization, 4-3 to 4-5
- domain, 4-4
- local, 4-4
- login, 4-4
- remote management, 3-8

V

- V.24 connection
 - baud rate, 3-11
 - device, 3-11
- VMS client
 - See* OpenVMS client
- VMSINSTAL, 9-4

W

- WAN Services for Windows NT
 - Core Services, 2-4
- Web server
 - integrating with Tomcat server, 6-2, 6-3, 6-5 to 6-6
 - receives SOAP messages, 6-3
 - security, 6-4
 - Tomcat, 6-2
- Windows 2000
 - server, 2-3
 - server installation, 2-1 to 2-12

- Windows 2000 client
 - See* Windows client
- Windows client
 - call processing API installation, 12-1 to 12-8
 - hardware required, 12-1
 - Windows Hardware Compatibility List*, 2-2, 12-2
- Windows NT
 - registry, 3-2
 - server, 2-3
 - server installation, 2-1 to 2-12
 - Windows NT Server, 2-3, 12-2
 - Windows NT Workstation, 2-3, 12-2
- Windows NT client
 - See* Windows client
- Windows XP
 - server, 2-3
 - server installation, 2-1 to 2-12
- Windows XP client
 - See* Windows client

X

- X.25 connection
 - device, 3-11, 3-14
 - LCN, 3-11
- XML format, 6-1