# ClarusIPC®

## Installation Guide

**Revision:** 2.5.0
**Date:** October 7, 2008

**CLARUS SYSTEMS®**

Clarus Systems, Inc.
2200 Bridge Parkway, Suite 101
Redwood City, CA 94065
http://www.clarussystems.com
Voice: 650-632-2800
Fax: 650-632-2810

*Copyright ©2008 Clarus Systems, Inc. All rights reserved. You cannot copy, reproduce, or duplicate any part of this document without written permission from Clarus Systems, Inc. One or more of the following trademarks may appear in this document. CallManager® is a registered trademark of Cisco Systems, Inc. ClarusIPC® is a registered trademark of Clarus Systems.*

# TABLE OF CONTENTS

# INSTALLING CLARUSIPC

This guide gets you up and running, and provides an overview of the system's user interface. It covers the following:

- System Requirements
- Installing ClarusIPC
- License Installation
- Server Group Setup

# System Requirements

This section provides the minimum and suggested hardware and software required to run ClarusIPC, and lists supported IP phones.

## Hardware

A server grade platform with the following recommended requirements:

- CPU: Pentium 4, 3.4 GHz Dual Core
- RAM: 4 GB
- Disk Space: 100 GB

## Software

- Operating system: Windows 2000, 2003, or 2008 Server

## Supported Cisco Phones

Please see the ClarusIPC Phone Model Support Matrix, included in the Appendices of the ClarusIPC User's Guide, for a full list of supported devices.

# Installing ClarusIPC

To install ClarusIPC, double-click the ClarusIPC installer to launch the InstallShield Wizard.
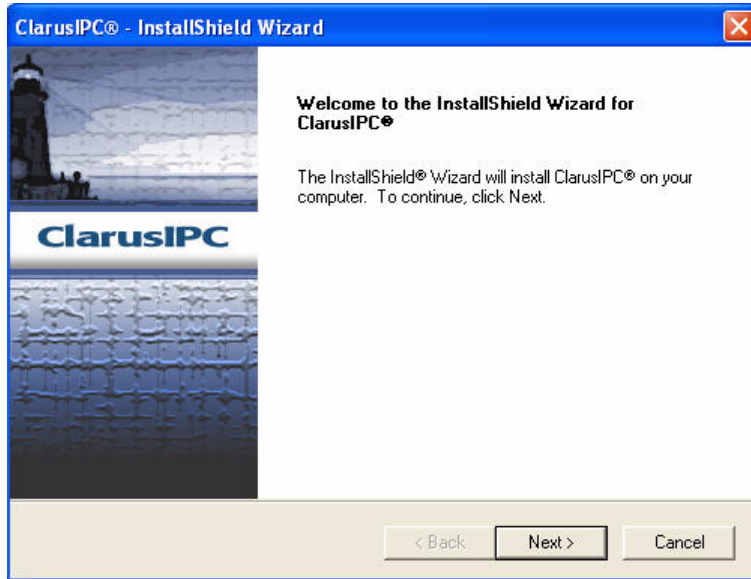


Figure 1  Run Installer

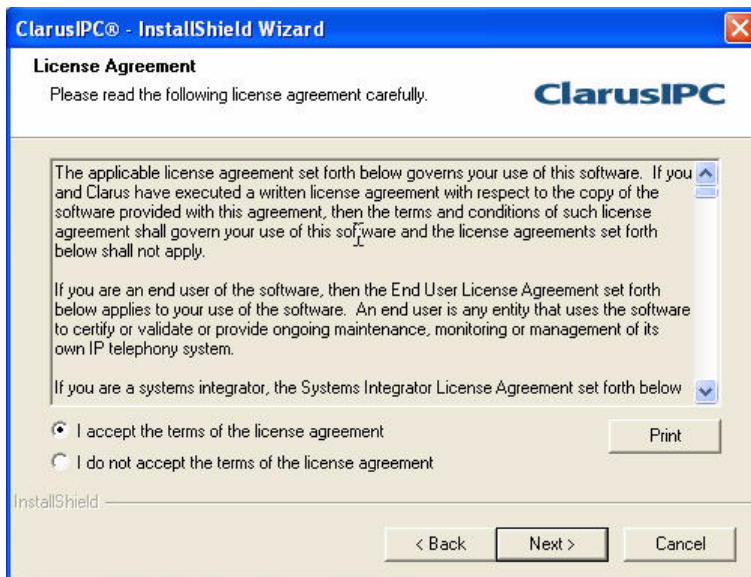1. Click **Next** to accept the License Agreement.



Figure 2  License Agreement

2. Accept the terms. Click **Next** to choose a destination folder.
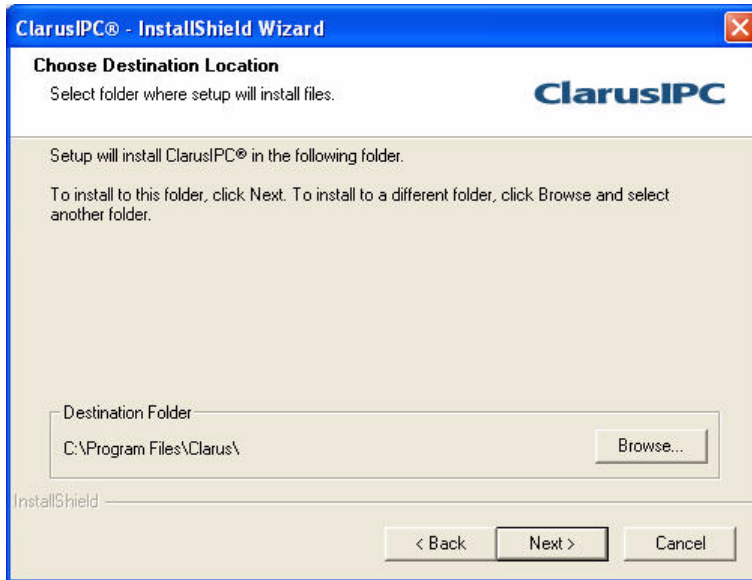


Figure 3  Destination

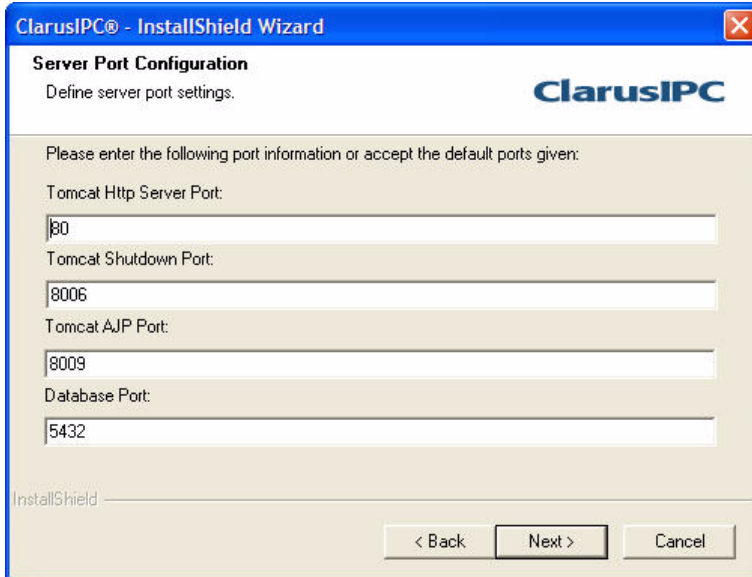3. Select a folder for the program files and click **Next** to configure the Server Ports.



Figure 4  Server Port Configuration

4. Accept the default values, or enter your own, and click **Next**. The Server Port Configuration process will query your system to determine that the ports you have defined are not already in use. The following progress window displays:



Figure 5  Validating Values

5. If there is a conflict an error screen will appear. Click **OK** and enter new values in the screen used in Figure 4.

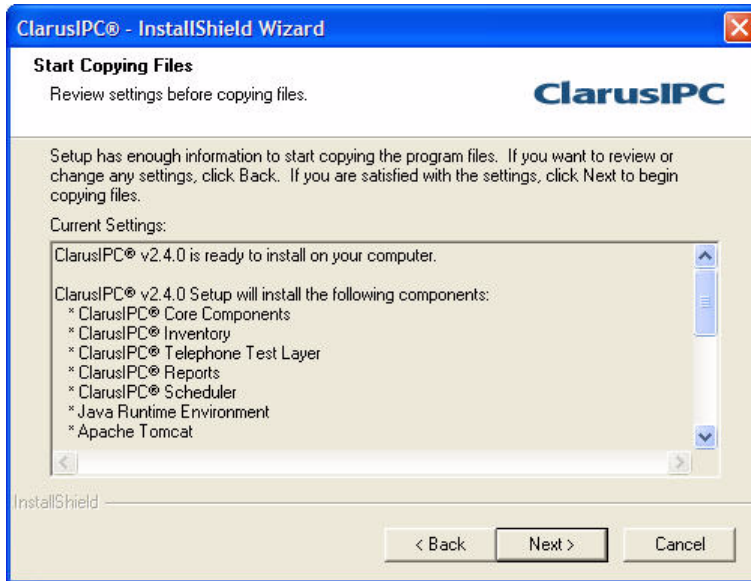6. Click **Next**. The following screen displays:



Figure 6  Copy ClarusIPC To Your Computer
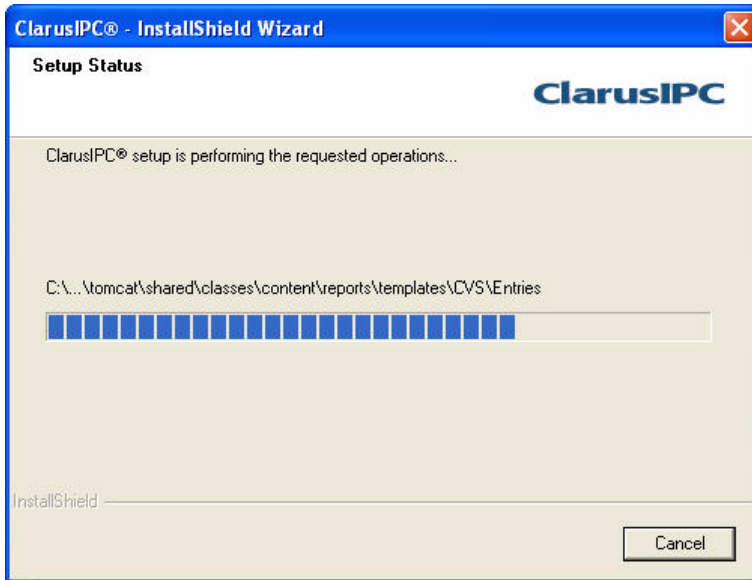
7. Click **Next** to begin installation.



Figure 7  Installation Status

8. When the Installer has finished, the following screen displays:
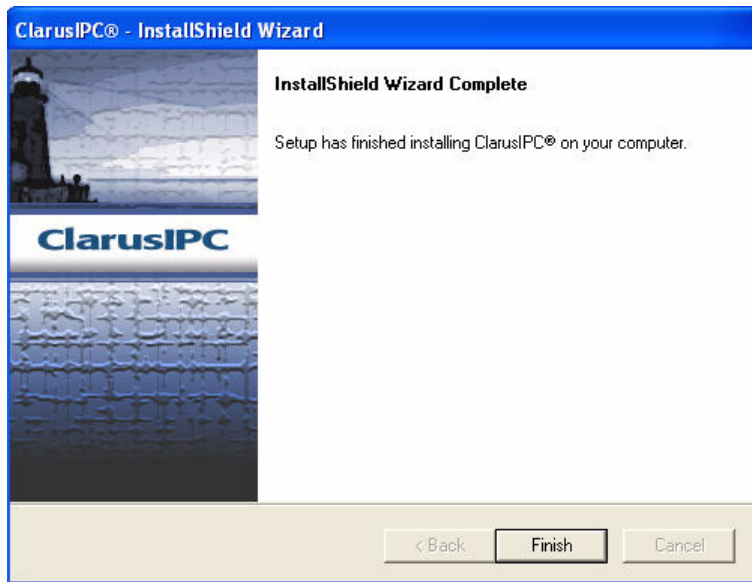


Figure 8  Installation Complete

9. Click **Finished**. You are now ready to run ClarusIPC. You do not need to restart the computer before using the product.

# License Installation

To enable ClarusIPC:

1.  Open a web browser to:

        http://<hostname or IP of ClarusIPC server>
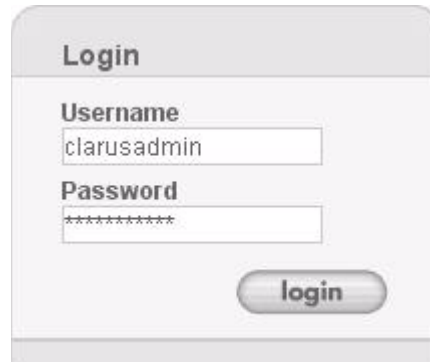
The following screen displays:



Figure 9  Login Window

2.  Enter **clarusadmin**, the default password **clarusadmin** and click **Login**. (To change the password, see the ClarusIPC User's Guide: *Getting Started*.)

3.  If this is the first time you have run the program, or if your license has expired, the following screen displays:



Figure 10  License Window

4.  Enter the license key given to you by Clarus Systems Technical Support, and click **save**. If you have not received a license key, click the **Request New License Key** button to generate an email to Clarus, requesting a new key.

5.  After entering the license key and clicking **save**, the main **Clusters** screen displays. You are now ready to use ClarusIPC.

Please note that your ClarusIPC installation is locked to the installed host, and cannot be moved to another machine without first contacting Clarus Systems.

See the *ClarusIPC User's Guide* for more information about running and using the product.

# ClarusIPC Server Groups

For environments in which multiple ClarusIPC tasks are simultaneously initiated, the amount of work performed in a given period puts a significant drain on the ClarusIPC server resources, ultimately reducing performance. To provide a more scaleable solution, ClarusIPC Server Groups (CSG) may be created. A CSG is a group of ClarusIPC servers which work together by leveraging a single, central database, to complete a set of tasks more efficiently. Work in the form of ClarusIPC tasks is dispatched to the most available ClarusIPC server in a load balancing fashion, dramatically improving overall performance of the system. Server Groups may be created to combine the resources of multiple servers working together.

ClarusIPC Server Groups (CSGs) provide scalability through clustering of multiple ClarusIPC servers, and include 3 Roles: DB Server, Controller, and Job Execution Servers (JES).

Any combination of functions may be co-resident on a single platform, but Clarus Systems recommends a minimum of 4 nodes in the group. All servers are intended to reside on the same 100Mbps (or better) LAN.

**NOTE:** ClarusIPC Server Groups is a licensable module, and is not part of any standard ClarusIPC package.

# Server Group Components

ClarusIPC Server Groups allow you to scale your ClarusIPC system to accommodate a higher workload, and add redundancy to the system.

A ClarusIPC Server Group consists of the following components:

- Database Server (one per Server Group),
- Controller (one per Server Group), and
- Job Execution Servers.

Each of these components may reside on the same server (standard deployment: no Server Group) or be distributed.

The **Database Server** maintains the single (Postgres) database server used by all Server Group members. This server is in charge of persisting all data, and retrieving it for requests. No replication of the data is included in this feature. There may be only one DB Server per server group.

The **Controller** runs Tomcat, manages all jobs, load balances, and dispatches work to the Job Execution Servers. If necessary, the Controller may be used as a Job Execution Server. There may be only one Controller per server group.

The **Controller** also:

- supports the ClarusIPC UI (navigation, add, edit, remove) for all users;
- controls Collector setup and Dashboard access;
- manages the ClarusIPC license;
- performs authentication via remote LDAP system;
- manages schedules and triggers task execution (local clock);
- performs device registration status collection to support Dashboard views;
- runs Voice Monitor rule processing.

**Job Execution Servers** run Tomcat, execute the jobs sent to them by the Controller, including CDR/CMR and KPI data collection, Test execution, Sync, and Staging, and return the results to the Database Server.

Multiple JES may be added to the group to help improve performance.

Both the Database Server and the Controller may perform some activities, but at a lesser priority than the Job Execution Servers. They are weighted such that only when the other Server Group members are fully loaded will they be asked to execute any jobs.

# Licensing

A single license is used for the entire ClarusIPC Server Group. ClarusIPC Licensing is performed only on the Controller, and node locking is enforced on the Controller. There are no restrictions to the number of Job Execution Servers that may join a Server Group via licensing.

# Server Group Setup

ClarusIPC Server Groups are based on a basic command and control model. This example creates a Server Group containing one Database Server, one Controller, and two Job Execution Servers.

Clarus recommends that all servers in the group be located within close network proximity, preferably on the same 100Mbps LAN.

**NOTE:** The clocks in all servers *must* be synchronized before ClarusIPC is used to execute jobs.

Servers should be configured in the order shown below.

## *Hardware recommendations*

Database recommendations

- **CPU:** 3.4 GHz dual/quad core Xeon processor
- **Disk:** RAID or equivalent disk array
    100 GB available storage.
- **RAM:** 4 GB

Controller and JES recommendations

- **CPU:** 3.4 GHz Xeon processor.
- **Controller and JES Disk:** 20 GB available storage
- **RAM:** 4 GB

## *Database Server Setup*

First, set up the Database Server.

1. Run the ClarusIPC installer; do not login or enter the license key.

2. From a cmdline, run the Server Group setup script from

    `%CLARUS_HOME%\tomcat\bin\sg_setup.bat`

3. Choose the option for "DB Server" and answer the questions as prompted.

(Do not access the ClarusIPC UI from this server.)

## *Controller Setup*

Second, set up the Controller.

1. Run ClarusIPC installer.

2. From a command line, run the Server Group setup script from

    `%CLARUS_HOME%\tomcat\bin\sg_setup.bat`

3. Choose the option for "Controller" and answer the questions as prompted.

4. Log into the UI via http://<Controller IP Address> and enter the license key.

*Job Execution Server Setup*

Finally, set up as many Job Execution Servers as necessary.

1. Run ClarusIPC installer.

2. From a command line, run the Server Group setup script from

   `%CLARUS_HOME%\tomcat\bin\sg_setup.bat`

3. Repeat for any additional JES servers.

When finished, log into the Controller via http://<Controller IP Address>, and select **status > server group**. Verify that all JES servers are Connected and Online.

*Add / Remove a Job Execution Server*

You may add a Job Execution Server to an existing ClarusIPC Server Group by modifying one configuration file on the Job Execution Server (clarusipc.properties) to point to the central Database Server and Controller (which may be on separate systems).

Although removing a Job Execution Server may require other Server Group activities to be halted, or the system brought down, a user may make a Job Execution Server unavailable (shut it down, stop the ClarusIPC services) and thereby take it out of service. When a Job Execution Server is taken out of service, new jobs are not sent to it. When the Job Execution Server becomes available again, new jobs may be sent to it without a system restart.

# Job Execution Server Status

The availability of a Job Execution Server may be checked from within the ClarusIPC UI from the **status > server group** menu item.

Jobs are dispatched to Job Execution Servers based on their availability and current load.

- Availability: An inactive Job Execution Server will always take precedence over a Job Execution Server running an active job.
- Current Load: The load on a Job Execution Server is defined by previous work sent that is still running. For example, a Test Plan weight is determined by the number of test components; and a Sync weight is determined by the estimated size using DN and Phone Counts from previous Syncs.

*Offline Job Execution Server*

If a Job Execution Server goes offline while processing a job, the Controller is notified that the job did not complete, and the job is moved into the waiting queue. If the job was part of a scheduled Task, it will be processed during its next scheduled execution. If the job was part of a user initiated event, such as a Sync or Test execution, the user must manually restart the operation.

# Publishing Reports

Reports are generated on the server that performs the work. Therefore, reports scheduled as part of a task may publish different URLs to access the reports. Tasks may also be run multiple times, on different Job Execution Servers. To consolidate all reports to a single location, or to consolidate reports for a single task, which has run multiple times over a given time span, use webdav to push the reports to a single server.

# Load Balancing

**Load balancing** schemes estimate the comparable weight of a task before determining where to send it. For example,

- Sync to 10k phone cluster is given a higher weight than a sync to a 1k phone cluster.
- A Test Plan containing 5000 components is given a higher weight than one containing 1000 components

All JES servers are assumed to have equivalent resources (CPU, Memory, etc).

When tasks are dispatched, they look at the work currently being performed by all JES to determine which is least loaded.

No Task is broken up when sent to a JES. Clarus Best Practices recommend making tasks as small as possible.

## *Help Desk*

Help Desk is handled differently. There is no native load balancing. Clarus recommends that you install separate ClarusIPC instances, and load balance among them. For these instances, the Cluster definition should use different CUCM servers in a Cluster, so as to distribute the load on RIS data collection. For more information, please contact your Clarus Systems representative.

# Server Failures

**If the DB Server fails** (or becomes unreachable by JES) while tasks are running:

- JES jobs will begin failing and the tasks will likely hang.
- Users cannot log into the Controller to stop Tasks.
- **Recovery:** stop all Tomcat services, start database service on DB Server, start Tomcat on Controller, start Tomcat on JES servers.

**If the Controller fails** (or becomes unreachable by JES) while tasks are running:

- Tasks will stop gracefully and not start the next operation in the task.
- No email notification will be sent.
- **Recovery:** Restart Tomcat on the Controller.

**If a JES becomes unavailable** while performing a task, the Controller will:

- Recognize that the task has not completed.
- Cancel the task that was dispatched. It will run at the next interval (if recurring) by an available JES.
- Any jobs that have completed will persist the results into the database.
- Task results will not be populated, and NO failure task email will be sent.

If a JES is unavailable when a task is dispatched, it will not be sent the task.

Upon recovery of the JES, it will immediately reregister with the Controller and be available to accept new tasks.

The Controller will serve as a JES *only* if there are no other JES available.
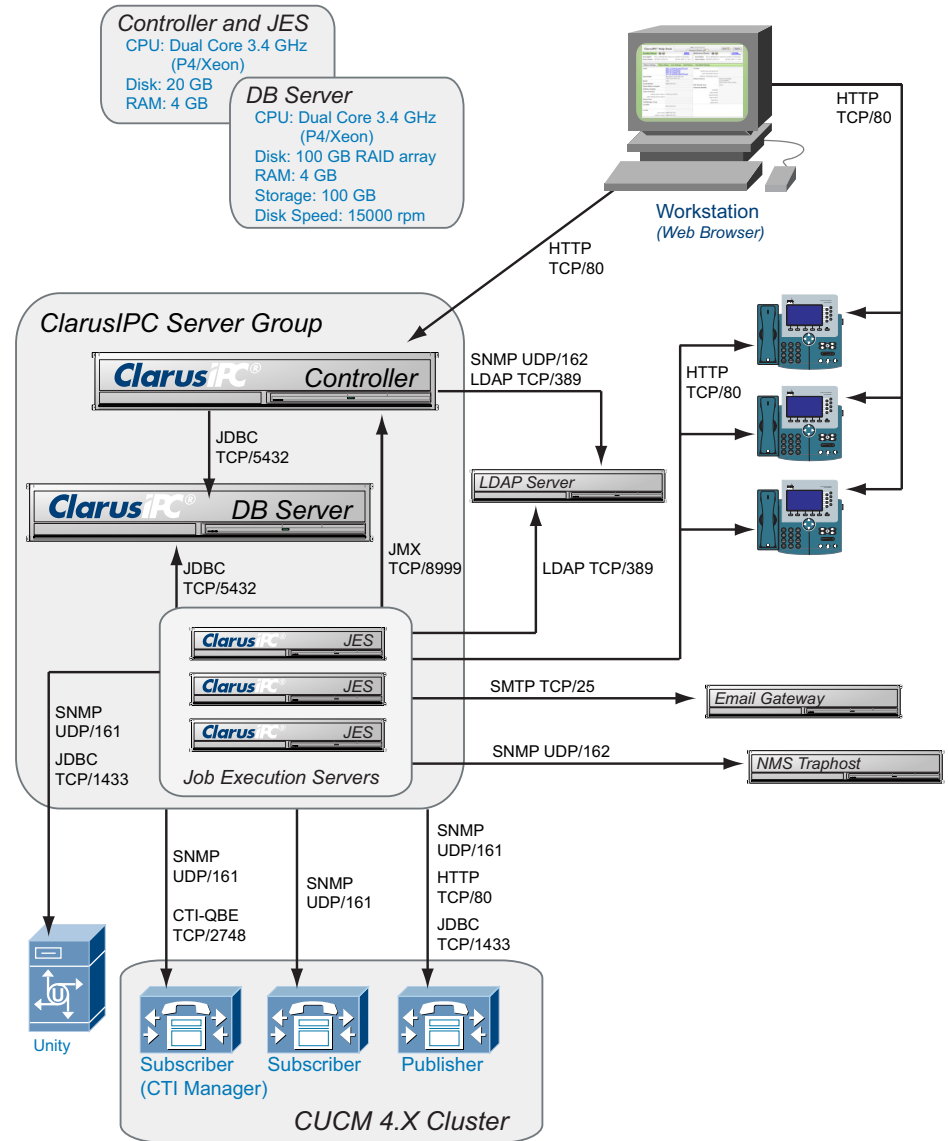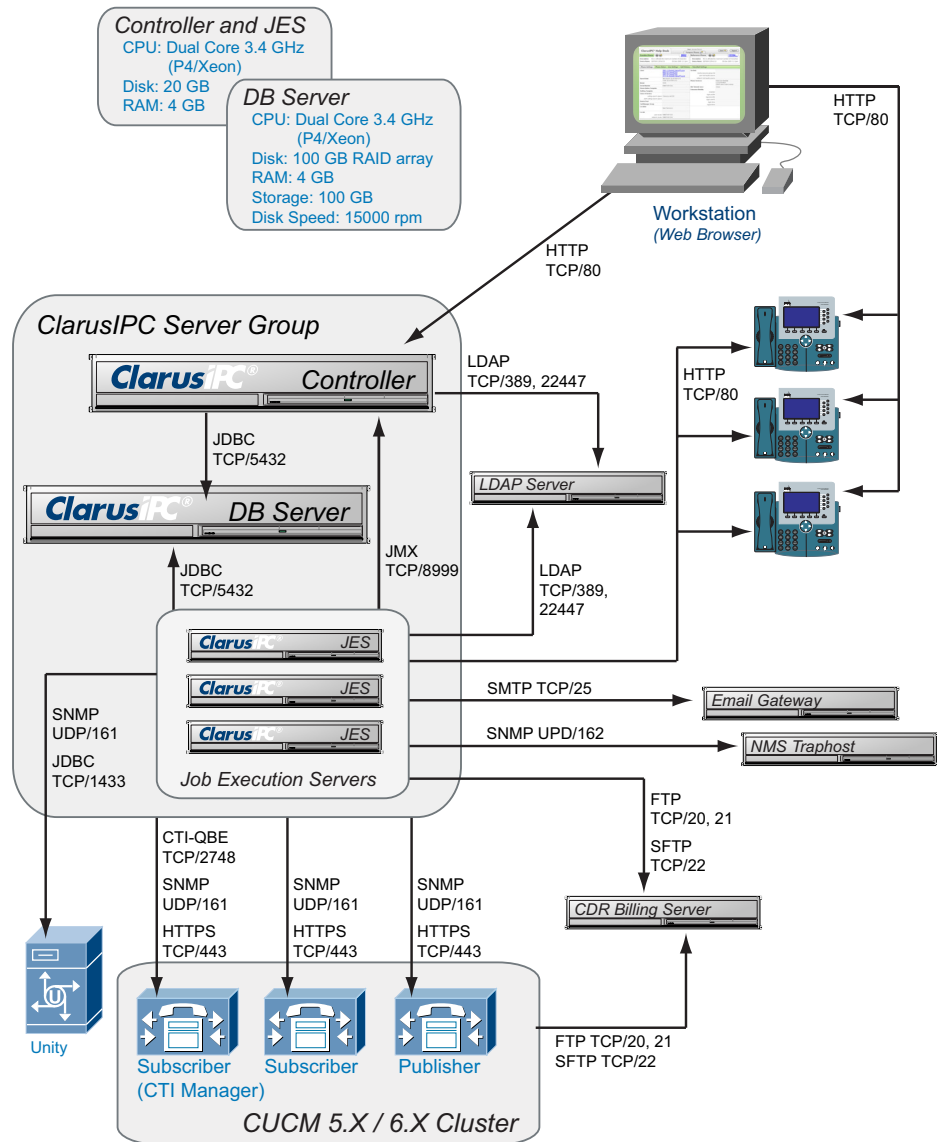


Figure 11  Server Group setup for CUCM 4.X Clusters

Figure 12  Server Group setup for CUCM 5.X and 6.X Clusters