



Expanded Meet-me Conferencing (EMMC) 1.0
Installation and Troubleshooting Guide
for the S8500

04-300527
Issue 4
March 2006

© 2006 Avaya Inc.
All Rights Reserved.

Notice

While reasonable efforts were made to ensure that the information in this document was complete and accurate at the time of printing, Avaya Inc. can assume no liability for any errors. Changes and corrections to the information in this document may be incorporated in future releases.

For full legal page information, please see the complete document, Avaya Legal Page for Software Documentation, Document number 03-600758.

To locate this document on the website, simply go to <http://www.avaya.com/support> and search for the document number in the search box.

Documentation disclaimer

Avaya Inc. is not responsible for any modifications, additions, or deletions to the original published version of this documentation unless such modifications, additions, or deletions were performed by Avaya. Customer and/or End User agree to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation to the extent made by the Customer or End User.

Link disclaimer

Avaya Inc. is not responsible for the contents or reliability of any linked Web sites referenced elsewhere within this documentation, and Avaya does not necessarily endorse the products, services, or information described or offered within them. We cannot guarantee that these links will work all of the time and we have no control over the availability of the linked pages.

Warranty

Avaya Inc. provides a limited warranty on this product. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya's standard warranty language, as well as information regarding support for this product, while under warranty, is available through the following Web site:

<http://www.avaya.com/support>

Copyright

Except where expressly stated otherwise, the Product is protected by copyright and other laws respecting proprietary rights. Unauthorized reproduction, transfer, and or use can be a criminal, as well as a civil, offense under the applicable law.

Avaya support

Avaya provides a telephone number for you to use to report problems or to ask questions about your product. The support telephone number is 1-800-242-2121 in the United States. For additional support telephone numbers, see the Avaya Web site:

<http://www.avaya.com/support>

Contents

Chapter 1: Overview.	5
Chapter 2: Installation.	7
Installation Overview	7
Installing EMMC	8
Installing the EMMC media server hardware.	8
Installing Software Disk 1 on the EMMC media server	8
Configuring Services on EMMC media server	9
Installing the EMMC Software.	15
Installing Software Disk 2	15
Installing the software from the web page	17
Final Steps	17
Verifying the Installation	17
Chapter 3: Upgrading EMMC	19
Overview	19
Process.	19
Appendix A: Configuring for EMMC on a Communication Manager media server	21
Detailed description of Expanded Meet-me Conferencing	21
Configuring for EMMC on a Communication Manager media server	21
Prerequisites for administering Expanded Meet-me Conferencing	22
Configuring Connectivity from Avaya Communication Manager 3.0 to the EMMC media server.	22
Configuring Announcements, Vectors, and Vector Directory Number (VDN) .	25
Change or create a VDN for Expanded Meet-me Conferencing:	27
Expanded Meet-me Conferencing display updates	29
Appendix B: Post Installation Configuration Changes	31
General System Configuration	31
Software Media Server Configuration	33
Proxy Configuration	33
Software Process Configuration	34
Appendix C: System Log Messages	37
Meeting Exchange Errors	37
Debug Process Errors.	37
SIP Utility Messages.	37
INIT Messages	40

Contents

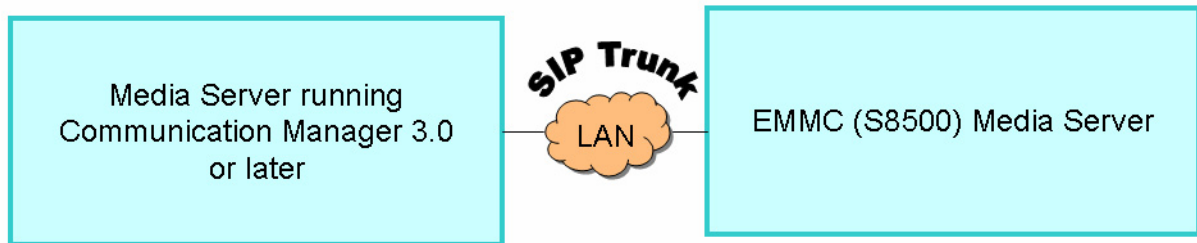
Appendix D: Troubleshooting	45
General Issues	45
Obtaining Debug Information	45
Restart a Program	46
Unable to create new files.	47
System Configuration Problems	47
System does not work with Dual SMP CPUs	47
Slow System Performance	47
System does not Accept VoIP Calls	48
VoIP programs restarted?.	49
Network Problems	49
Debug a Network Problem	49
Network is unreachable	50
Network is unreachable: Firewall.	51
How can I capture or monitor IP traffic?	51
Audio Problems	51
No Audio	51
Debug Process.	51
Error Messages	52
SIP Utility Errors	52
Appendix E: Utilities.	59
Introduction	59
Diagnostic Data for Meeting Exchange	59
Using ipinfo	59
Syntax	60
Lock file	61
Configuration Files	61
Bridge Properties	62
Conference Properties.	62
Meeting Properties.	62
Meeting List Properties	63
Line Properties.	63
Participant Properties	63
Index	65

Chapter 1: Overview

Expanded Meet-me Conferencing (EMMC) is a conferencing application that runs on an Avaya S8500 Media Server, referred to as the EMMC server. EMMC interfaces with an Avaya Communication Manager system version 3.0 or later on a separate Linux-based server, referred to as the Communication Manager server.

The EMMC server connects to (over the network) and works with an Avaya Communication Manager server, which is different than the EMMC server. The version of software running on the Communication Manager server does not need to match that of the base Communication Manager software installed on the EMMC server.

Figure 1: Replace variable w/ short product name connection to Communication Manger



EMMC enhances the Avaya Communication Manager six-party Meet-Me conference feature, expanding the number of available ports to 300. EMMC provides external conferences based on a SIP trunk interface to a conference server, which acts as the audio conference device.

EMMC has the following capabilities:

- 300 ports of G.711 A-law or μ -law
- RFC 2833 DTMF support
- In-band DTMF support
- Up to 300 participants in a single conference
- Full support of the media server interface

Overview

Chapter 2: Installation

This chapter overviews the installation procedures for the 8500 media server running Meet-me Conferencing.

Installation Overview

The EMMC application runs as a feature of Avaya Communication Manger. The Communication Manager software runs on one of the Avaya Linux-based media servers: S8300, S8500, or S8700-series. The EMMC software runs on a separate Avaya S8500B/C (IBM 306) Media Server.

The software required to run EMMC is distributed on two CDs labeled EMMC Software Disk 1 and EMMC Software Disk 2. Disk 1 contains Avaya Communication Manager software and Disk 2 contains EMMC software. Both sets of software are installed on the EMMC media server. As part of the EMMC software installation, Communication Manger is disabled on the EMMC media server.

This document assumes that you are adding EMMC to an existing Avaya communications system running Communication Manager, release 3.0 or later.

The major steps to install EMMC are:

1. Install the EMMC hardware (S8500) using *Quick Start for Hardware Installation: S8500 Media Server*.
2. Install EMMC Software Disk 1 on the EMMC media server using *Installing and Configuring the S8500 Media Server*.

 **Important:**

If there is a SAMP firmware update, install it at this time.

3. Configure Services on the EMMC media server. See [Installing the EMMC media server hardware](#) on page 8
4. Use EMMC Software Disk 2 to install EMMC software on the EMMC media server. See, [Installing the EMMC Software](#) on page 15
5. Configure EMMC on the Communication Manager media server, if not already completed. Refer to [Appendix A: Configuring for EMMC on a Communication Manager media server](#) on page 21.

Installing EMMC

This section describes the procedures for adding EMMC to an existing Linux-based communication system.

Installing the EMMC media server hardware

 **Important:**

If there is a SAMP update required, this must be done BEFORE installing the EMMC Software.

To install the S8500 hardware follow the instructions in *Quick Start for Hardware Installation: S8500 Media Server* up to “Connecting to the IPSIs.”

Note:

In addition to the inventory listed in the *Quick Start*, there will be two Installation CDs.

Configure the SNMP module in the UPS. Follow the instructions in *Installing and Configuring Avaya S8500 Media Server*.

Installing Software Disk 1 on the EMMC media server

To install Communication Manager software on the EMMC media server, complete the following sections in Chapter 2 of *Installing and Configuring Avaya S8500 Media Server*.

1. “Clearing the ARP cache on the laptop”
2. “Powering up the media server”
Use the EMMC Software Disk 1 CD.
3. “Accessing the media server”
4. “Configuring Telnet for Windows XP/2000”
5. “Installing Avaya Communication Manager”

 **Important:**

If there is a SAMP firmware update, install it at this time. If there is a post-installation service pack for this release of Communication Manager, install it at this time.

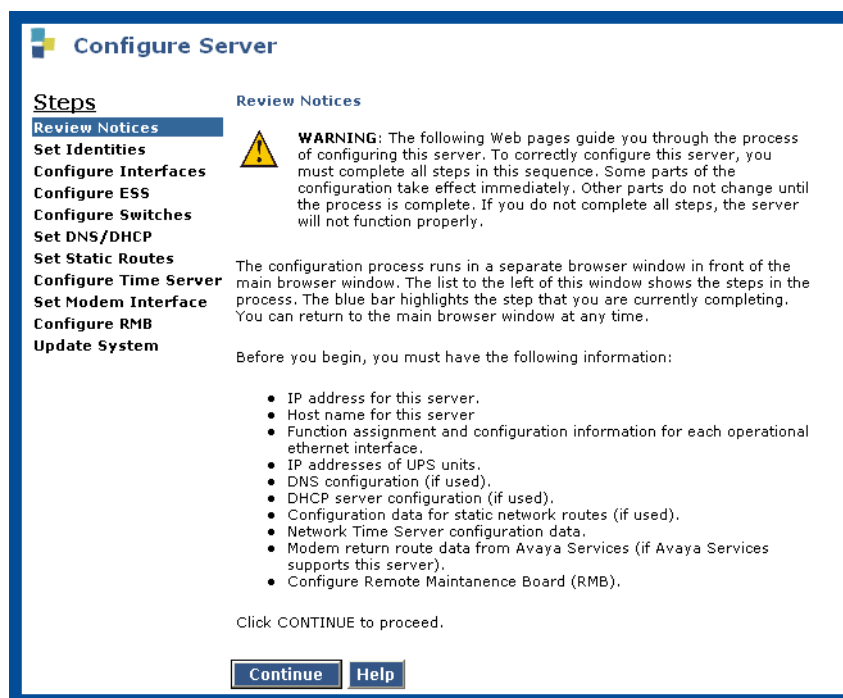
Configuring Services on EMMC media server

Important:

Do not use the Avaya Installation Wizard. Do not start translation of the Communication Manager application that is now running on the EMMC server.

To configure Services, complete these steps:

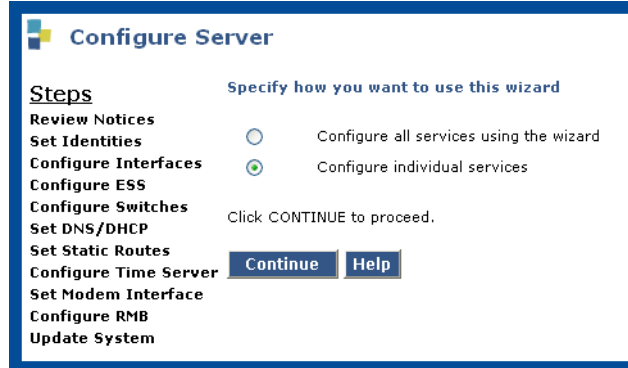
1. Using a web browser, connect to the EMMC media server and log in.
2. Click **Launch Maintenance Web Interface**.
3. Click **Server Date/Time** under Server.
Enter the current date and time and the local time zone.
4. Click **Configure Server** located in the set of commands available on that web page. The system displays the **Configure Server** web page.



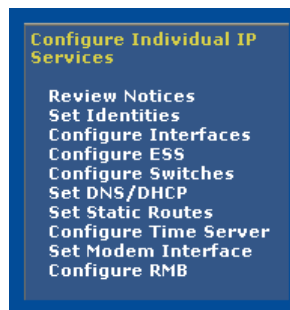
5. Click **Continue**. The system prompts you to backup data before continuing. If this is a new installation, you do not need to back up any data.

Installation

6. Click **Continue**. The system displays the following web page.



7. Select **Configure Individual Services** and click **Continue**. The system notifies you that you can use the links on the left of the page to configure IP services and displays the Configure Individual Services menu show here.



8. Select the **Set Identities** option from the Configure Individual Services menu. The system displays the **Set Server Identities** web page.

Configure Server

Set Server Identities

The host name and ID of each server must be unique.

Select NIC Usage

Host Name ID
(Range: 1 to 99)

Indicate how each ethernet port is to be used. You may accept the defaults. Ethernet ports may be used for multiple purposes, except for the port assigned to the laptop, which must be dedicated to only that purpose. Physical connections to the Ethernet ports must match these settings.

1. Control Network A (Default: Ethernet 0)

2. Services Port (Default: Ethernet 1)

3. Control Network B (Default: Ethernet 2)

4. Corporate LAN (Default: Ethernet 3)

Click CONTINUE to proceed.

- Enter the **EMMC** host name.
 - Enter the **NIC usage** (use the default setting)
 - Click **Change**.
 - Click **Close Window**.
9. Select **Configure Interfaces** from the Configure Individual Services menu. The system displays the **Configure Ethernet Interfaces** web page.

Configure Server

Configure Ethernet Interfaces

Ethernet 0: Control Network A And Corporate LAN Interface

IP address server1 (avaya306-2)

Gateway

Subnet mask

Speed (Current speed : 100 Megabit full duplex)

Enable VLAN 802.1q priority tagging

Ethernet 1: Laptop

IP address

Subnet mask

Click CHANGE to change values.

Installation

a. Enter the customer provided values for Ethernet 0, including the:

- **EMMC Server IP address**
- **Subnet mask**
- **Gateway address**
- **Speed**

Note:

The **Speed** setting must match the Ethernet switch's port setting.

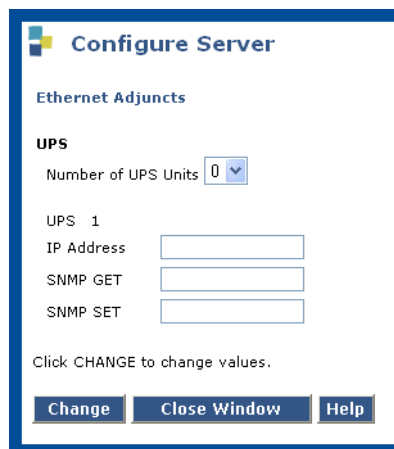
b. Click **Change**.

10. Select **Configure ESS** from the Configure Individual Services menu.

11. Select "**this is not an ESS**".

12. Click **Change**.

13. Configure **Switches** from the Configure Individual Services menu. The system displays the Ethernet Adjuncts web page.



Configure Server

Ethernet Adjuncts

UPS

Number of UPS Units

UPS 1

IP Address

SNMP GET

SNMP SET

Click CHANGE to change values.

a. Enter the number of **UPS** and customer provided **SNMP** information, if applicable.

b. Click **Change**.

14. If applicable select **Set DNS / DHCP** from the Configure Individual IP Services menu. The system displays the Configure External DNS Server Configuration web page.

The screenshot shows a web form titled "Configure Server" with a sub-section "External DNS Server Configuration". A note states: "Note: If DNS is not used, leave these fields blank." Under "Name Servers", there are three "IP Address" fields (1, 2, 3) and one "DNS Domain" field. Below these are five "Search Domain" fields. A section titled "DHCP Service on this Server:" contains a note: "(This server may not function properly if this option is set incorrectly. See Help for details.)" and a checkbox labeled "Enable DHCP service on this server for IPSIs". A red circle highlights this checkbox, and a red arrow points from a note on the left to it. At the bottom of the form are three buttons: "Change", "Close Window", and "Help".

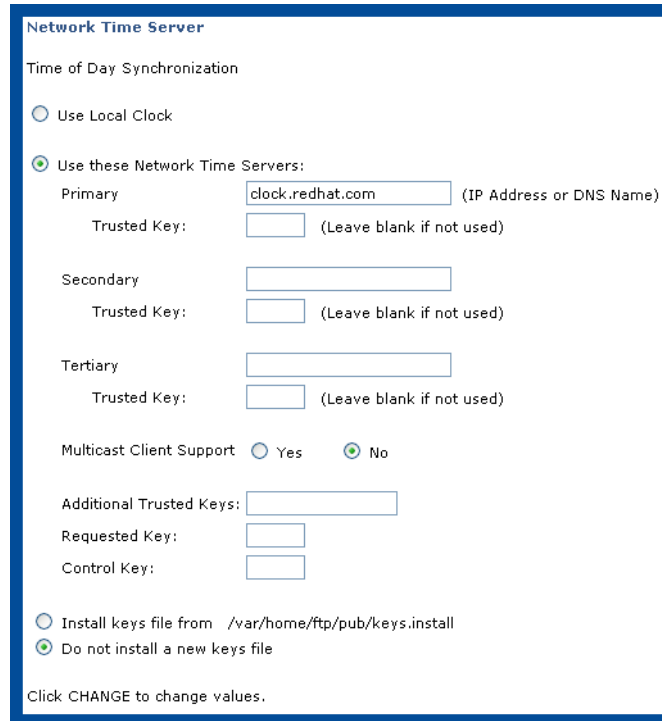
Note:

Do not select **Enable DHCP service on this server for IPSIs**.

- Enter any customer required values, such as **Domain Name server** and **DHCP server** information.
- Click **Change**.

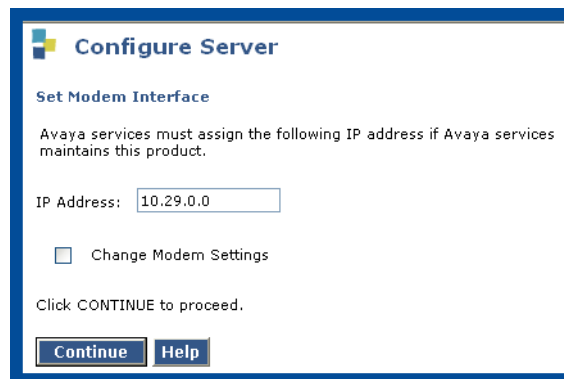
Installation

15. Optionally, select **Configure Time Server** from the Configure Individual IP Services menu. The system displays the Network Time Server configuration web page.



The screenshot shows the "Network Time Server" configuration page. It is titled "Network Time Server" and has a sub-heading "Time of Day Synchronization". There are two radio buttons: "Use Local Clock" (unselected) and "Use these Network Time Servers:" (selected). Under "Use these Network Time Servers:", there are three sections: "Primary", "Secondary", and "Tertiary". Each section has a text input field for the server name (with "(IP Address or DNS Name)" next to the Primary field) and a "Trusted Key:" label with an empty text input field and the instruction "(Leave blank if not used)". Below these is a "Multicast Client Support" section with "Yes" and "No" radio buttons, where "No" is selected. There are also three more text input fields: "Additional Trusted Keys:", "Requested Key:", and "Control Key:". At the bottom, there are two radio buttons: "Install keys file from /var/home/ftp/pub/keys.install" (unselected) and "Do not install a new keys file" (selected). A note at the bottom says "Click CHANGE to change values."

- a. Enter any customer provided required values, such as **Time Server**.
- b. Click **Change**.
16. Optionally, select **Modem Interface** from the Configure Individual IP Services menu. The system displays the web page.



The screenshot shows the "Configure Server" web page. It has a sub-heading "Set Modem Interface". Below this, there is a note: "Avaya services must assign the following IP address if Avaya services maintains this product." There is a text input field for "IP Address:" containing "10.29.0.0". Below that is a checkbox labeled "Change Modem Settings" which is unchecked. At the bottom, there is a note: "Click CONTINUE to proceed." and two buttons: "Continue" and "Help".

17. Select **Configure RMB** from the Configure Individual IP Services menu and verify the information.

RMB Network Configuration	
Configure Remote Maintenance Board (RMB)	
Services Laptop:	
LAN IP Address	192.11.13.6
Gateway IP Address	0.0.0.0
Subnet Mask	255.255.255.252
Reserved (Services Future Use):	
LAN IP Address	<input type="text" value="192.11.13.6"/>
Gateway IP Address	<input type="text" value="0.0.0.0"/>
Subnet Mask	<input type="text" value="255.255.255.254"/>

18. Return to a Linux command prompt and set the Product IDs:

- a. To set the SAMP(RMB):

```
productid -r 5XXXXXXXXX
```

- b. To set the CM:

```
productid -p 1XXXXXXXXX
```

Installing the EMMC Software

The EMMC media server software may be installed either from the EMMC Software Disk 2 or from the Maintenance Web Interface after the files have been downloaded to the server. The two methods are the same except for the first few steps.

The EMMC software is provided in the .tar.gz file in the format of a Communication Manager service pack (software update file). The file updates a particular version of Communication Manager software. You can identify the CM version that the file updates because the version is included in the name of the .tar.gz file before the “-emmc”.

The Communication Manager release on the Communication Manager media server does NOT need to match that of the Communication Manager software installed on the EMMC server.

Installing Software Disk 2

Follow this procedure when using a CD to install the EMMC Media Server Software:

1. Insert Software Disk 2 into the installer’s laptop computer.
2. Connect to the services port of the EMMC server from the laptop.
3. Open a browser and log in and browse to the “**Download Files**” maintenance web page.

Installation

4. Select "**Files to download from the machine I'm using to connect to the server.**" The system displays a web page.
5. Select "**Browse**" to locate the CD and select the update file (**.tar.gz*).

Note:

You may not see the file if Windows Explorer is set to hide files.

6. Download the update file.
7. Use Telnet or SSH to access the EMMC media server (192.11.13.6) and log in.
8. Run the update_unpack command on the *.tar.gz file:

```
# update_unpack update-file-name.tar.gz
```

where *update-file-name* is the name of the downloaded file. The file name format is:

```
CMversion-emmc.EMMCversion.tar.gz
```

where *CMversion* identifies the Conference Manager version and *EMMCversion* identifies the EMMC version.

9. Run the update_activate Linux command on the newly unpacked update file.

```
# update_activate CMversion-emmc.EMMCversion
```

10. Answer questions displayed by the system:

```
"Do you want to create a login account for user "cust" ?  
"Do you want to create a login account for user "dadmin" ?  
"Enter a value for DiffServeTOSValue" Accept the default.  
"Enter a value for EthernetVlanValue" Accept the default.  
"Enter a value for AutomaticGain" Accept the default.  
"Enter a value for ComfortNoiseGeneration" Accept the default.
```

Once the conference bridge software restarts, the EMMC software is installed.

Note:

Do not interrupt the file load process: when the process completes, the system displays a prompt.

11. Confirm that the bridge is running by running this command:

```
service ipcb status
```

The system will report "ipcb is running".

12. Follow the directions in [Appendix A: Configuring for EMMC on a Communication Manager media server](#) on page 21.

Installing the software from the web page

Use this procedure only if you did not install the EMMC software from the CD.

When installing the EMMC Media Server Software from the maintenance web page, follow this procedure:

1. Open a browser using one of these techniques:
 - Connect to the services port from a laptop and use IP address 192.11.13.6.
 - When connected to the customer's LAN, use the IP address configured in [Configuring Services on EMMC media server](#) on page 9.
2. Log in and browse to the "Download Files" maintenance web page.
3. Select **File(s) to download from the LAN using URL** and enter a URL where the update file can be found (*.tar.gz).
4. Download the update file.
5. Continue with steps 7 through 12 in the previous procedure for [Installing Software Disk 2](#).

Final Steps

Installers need to complete these steps:

- Test Conference bridge from different endpoint (i.e. dcp,sip,h.323)
- Load Authentication file on EMMC

Verifying the Installation

During installation the system records three log files. These files may be used to troubleshoot a problematic installation. The log files are:

- ipcb_cdinst.sh.log
- ipcb_cdinst.sh_verbose.log
- messages.log

The system records a summary of the installation process in /usr/ipcb/ipcb_cdinst.sh.log. The log file shows the status of each major installation step. It is useful for verifying the status of the installation.

The system records a second log file that is useful when there is a problem with the installation. This file /usr/ipcb/ipcb_cdinst.sh_verbose.log contains a complete list of everything that displayed on the screen during installation.

Installation

The installer writes all major events and status results to `/var/log/messages`. The system log is chronological so it displays other events that occurred during the installation. It also logs any entry made to the command lines by the root user who ran the installer.

There can be more than one messages file in the `/var/log` directory because Linux rotates log files (backup old, large one). The message file for install will correspond to the EMMC installation date and time.

Chapter 3: Upgrading EMMC

EMMC configurations upgrading from Communication Manager (CM) 3.0 or 3.0.1 to CM 3.1 and later releases on the media server need to upgrade the EMMC release on the EMMC server.

Note:

EMMC 1.0.17 enables EMMC to operate with both CM 3.0, 3.0.1, CM 3.1, and later releases.

Overview

Upgrading the EMMC is the same as installing a new service pack. Refer to the S8500 portion of the Software Update Instructions for S8300, S8500, and S8700/S8710 Media Servers Running Communication Manager 2.0 and Later Releases.

To access this document, refer to the [Avaya Support Downloads](#) and then click the link titled, "Communication Manager 2.0 and Later Software Update Procedures "

Process

Prior to installing an upgrade of Communication Manager 3.1 and later releases on the Communication Manager Server, you must upgrade the EMMC software on the EMMC server.

Follow this procedure to upgrade the EMMC software on the EMMC server.

1. Copy the EMMC update from the EMMC software CD onto the EMMC server.
2. Unpack the update.
3. Deactivate the version of EMMC currently activated and running.
4. Activate the new EMMC update.



Important:

The software on the EMMC Server continues to be the CM 3.0 load 340.3.

Upgrading EMMC

Appendix A: Configuring for EMMC on a Communication Manager media server

Use the Expanded Meet-me Conferencing (EMMC) feature to set up multi-party conferences consisting of more than six parties. The Expanded Meet-me Conferencing feature supports up to 300 parties.

For more information on Expanded Meet-me Conferencing, see *the Feature Description and Implementation for Communication Manager, 555-245-205 Issue 3, April 2005*.

Detailed description of Expanded Meet-me Conferencing

This section provides a detailed description of the Expanded Meet-me Conferencing feature.

The Expanded Meet-me Conferencing feature is an enhancement to the Avaya Communication Manager six-party Meet-me Conference feature. The Expanded Meet-me Conferencing feature supports up to 300 parties.

The Expanded Meet-me Conferencing feature is supported on all Linux platforms of Avaya Communication Manager (S8700, S8500, S8300) running Avaya Communication Manager release 3.0 software or higher.

Note:

For more information on Expanded Meet-me Conferencing, refer to *SIP Support in Avaya Communication Manager 3.0, 555-245-206, Issue 3, May 2005*.

Configuring for EMMC on a Communication Manager media server

After installing the Replace variable w/ short product name software, configure the initial settings by visiting the Communication Manager Maintenance Web Interface. These pages are described in the Screen Reference chapter of the *Administrator Guide for Avaya Communication Manager – 03-300509, Issue 1, April 2005*. and Expanded Meet-me Conferencing settings are annotated in this section.

Prerequisites for administering Expanded Meet-me Conferencing

You must complete the following actions before you can administer the Expanded Meet-me Conferencing feature:

- On the **Optional Features** screen, ensure that the **G3 Version** field is set to **V13** or later. If the **G3 Version** field is not set to **V13** or later, your system is not enabled for the Expanded Meet-me Conferencing feature. Contact your Avaya representative before you continue with this procedure.

To view the **Optional Features** screen, type `display system-parameters customer-options`. Press **Enter**.

- On the **Optional Features** screen, click **Next** until you see the **Enhanced Conferencing** field. Ensure that this field is set to **y**. If the **Enhanced Conferencing** field is set to **n**, your system is not enabled for the Expanded Meet-me Conferencing feature. Contact your Avaya representative before you continue with this procedure.
- On the **Optional Features** screen, a numeric value must appear on the **Maximum Number of Expanded Meet-me Conference Ports** field. The maximum value for this field is 300, and it corresponds to the number of Expanded Meet-me Conferencing ports that your system has. This number can be less than 300, and is determined by the license file for your system.
- On the **Optional Features** screen, the following fields must also be set to a non-zero value:
 - **Maximum Administered SIP Trunks**
 - **Maximum TN2501 VAL Boards** *or* **Maximum G250/G350/G700 VAL Sources**
- On the **Optional Features** screen, the following fields must also be set to **Y**:
 - **ISDN-PRI**
 - **IP Trunks**

Configuring Connectivity from Avaya Communication Manager 3.0 to the EMMC media server

Follow these steps to configure connectivity from Avaya Communication Manager to the Meeting Exchange.

1. Customer must verify that they have:
 - a. a codec set that supports only G711 and does not enable media encryption.
 - b. an IP network region that supports the codec set that is G711 only.

2. Set the inter-regions' codec-set to that specified in 1a. This is set on page 3 of change network-region ip .
3. Add a Node Name for the EMMC server by entering **change node-name IP**.
4. Select a CLAN or G700 for connection to the EMMC. Note its node-name.
5. Add one or two signaling groups. For each Signaling Group:
 - a. Set the:
 - **Group Type** to "sip"
 - **Near-end Node Name** to value set in step 4, i.e. "CLAN"
 - **Far-end Node Name** to the same setting used in step 3, i.e.EMMC.
 - **Far-end Listen Port** to "5061"

Note:

The setting, "5061" is required for the **Far-end Listen Port**.

- **Far-end Network Region** to the codec set referred to in step 1.
- b. Verify the **Transport Method** is set to "TLS"

Figure 2: Signaling Group screen

```
SIGNALING GROUP

Group Number: 2                Group Type: sip
                                Transport Method: tls

Near-end Node Name: CLAN        Far-end Node Name: EMMC
Near-end Listen Port: 5061      Far-end Listen Port: 5061
Far-end Domain:                 Far-end Network Region: 1

                                Bypass If IP Threshold Exceeded? n

DTMF over IP: rtp-payload       Direct IP-IP Audio Connections? y
                                IP Audio Hairpinning? y

Session Establishment Timer(min): 120
```

6. Add one or two Trunk groups and set the **Number of Members** for each group.

Configuring for EMMC on a Communication Manager media server

Note:

If the Number of Members is set to "0", you will not be able to administer the members for this trunk group correctly.

Note:

The total number of members should match the Maximum number of Expanded Meet-me Conference Ports in the Optional Features. For example, if the Maximum Number of Ports is 300, then the number of members set in the first trunk group can be 255, and the number of members in set in the second trunk group can be 45. (Maximum number of members per trunk group is 255).

Figure 3: Trunk Group screen

```
TRUNK GROUP
Group Number: 2                Group Type: sip                CDR Reports: y
  Group Name: To EMMC          COR: 1                TN: 1                TAC: 702
  Direction: two-way          Outgoing Display? n
  Dial Access? n              Busy Threshold: 255        Night Service:
  Queue Length: 0
  Service Type: tie           Auth Code? n
                                Signaling Group: 2
                                Number of Members: 10

TRUNK PARAMETERS
  Unicode Name? y
                                Redirect On OPTIM Failure: 5000
  SCCAN?n                     Digital Loss Group: 18
-----
TRUNK FEATURES
  ACA Assignment? n           Measured: none
                                Maintenance Tests? y
                                Numbering Format: public
                                Replace Unavailable Numbers? n
```

Note:

The system needs 1 or 2 signaling groups and 1 or 2 trunk groups. The combined Number of Members from both trunk groups must be less than or equal to the number of ports of EMMC licensed on the system.

7. Add a **Route Pattern** for the Trunk Groups. For additional information, see *Feature Description and Implementation for Avaya Communication Manager 555 245 205, Issue 3 June 2005*.

8. Modify the **AAR analysis** with the trunk information. For additional information, see *Administrator Guide for Avaya Communication Manager 03-300509, Issue 1 June 2005*.

Note:

After this step, the Installer should be able to establish a talk path by dialing a route-to number.

Configuring Announcements, Vectors, and Vector Directory Number (VDN)

Create the necessary announcements. Some sample announcements are:

- EMMC_Welcome_Code_Prompt
`Welcome to your conference bridge, please enter your password now.`
- EMMC_Code_Invalid_Try_Again
`Your password is invalid; please try again.`
- EMMC_Code_Invalid_Goodbye
`Your password is invalid. Goodbye.`
- EMMC_Sorry_Full
`We're sorry, your conference is full. Please contact your administrator.`
- EMMC_Join_First
`Welcome to your conference; you are the first to join.`
- EMMC_Join_In-Progress
`Welcome. Your conference is already in- progress.`
- EMMC_Problems_Connecting
`Your call cannot be routed at this time. Please try again later.`

Configuring for EMMC on a Communication Manager media server

Use the **Announcements/Audio sources** screen to set the announcements for the EMMC.

Figure 4: Announcements /Audio Sources Screen

```
change announcements                                     Page 1 of 16
ANNOUNCEMENTS/AUDIO SOURCES
```

Ann. No.	Ext.	Type	COR	TN	Name	Q	QLen	Pr	Rt	Port	Group/
1	6001	integrated	1	1	DNC_Call_Blocked	y	NA	n	64	001V9	
2	6002	integrated	1	1	DNC_Bad_Override	y	NA	n	64	001V9	
3	6003	integrated	1	1	DNC_Error	y	NA	n	64	001V9	
4	6004	<i>integrated</i>	1	1	<i>EMMC_Welcome_Code_Prompt</i>	n	NA	n	64	001V9	
5	6005	<i>integrated</i>	1	1	<i>EMMC_Code_Invalid_Try_Again</i>	n	NA	n	64	001V9	
6	6006	<i>integrated</i>	1	1	<i>EMMC_Code_Invalid_Goodbye</i>	n	NA	n	64	001V9	
7	6007	<i>integrated</i>	1	1	<i>EMMC_Sorry_Full</i>	n	NA	n	64	001V9	
8	6008	<i>integrated</i>	1	1	<i>EMMC_Join_First</i>	n	NA	n	64	001V9	
9	6009	<i>integrated</i>	1	1	<i>EMMC_Join_In-Progress</i>	n	NA	n	64	001V9	
10	6010	<i>integrated</i>	1	1	<i>EMMC_Problems_Connecting</i>	n	NA	n	64	001V9	

9. Create an Expanded-meet Me vector:

```
change vector xxxxx
```

where x is the vector number

Figure 5: Call Vector screen

```

CALL VECTOR

Number: 9                      Name: Expanded MeetMe
Multimedia? n                 Attendant Vectoring? n     Meet-me Conf? y             Lock? y
Basic? y                      EAS? y      G3V4 Enhanced? y         ANI/II-Digits? y          ASAI Routing? y
Prompting? y                  LAI? n     G3V4 Adv Route? y        CINFO? y      BSR? y           Holidays? y
Variables? y                  3.0 Enhanced? n

01 collect                    6          digits after announcement 6004
02 goto step                  6          if digits                  =          meet-me-access
03 collect                    6          digits after announcement 6005
04 goto step                  6          if digits                  =          meet-me-access
05 disconnect                 after announcement 6006
06 goto step                  12         if meet-me-full
07 goto step                  14         if meet-me-idle
08 announcement               6009
09 route-to                   meetme
10 disconnect                 after announcement 6010
11 stop

change vector 9

CALL VECTOR

12 disconnect                 after announcement 6007
13 stop
14 announcement               6008
15 goto step                   9          if unconditionally
16
17
18

```

Note:

This vector may be used for both standard meet-me VDN's and expanded meet-me VDN's.

Change or create a VDN for Expanded Meet-me Conferencing:

1. Type `change vdn xxxxx`, where `xxxxx` is the vector directory extension number. The system displays the [Vector Directory Number screen](#) like the one in [Figure 6](#).

Figure 6: Vector Directory Number screen

```
change vdn 50300                                     Page 1 of 3
                                                    VECTOR DIRECTORY NUMBER
                                                    Extension: 50300
                                                    Name: EMMC VDN
                                                    Vector Number: 10
                                                    Meet-me Conferencing? y
                                                    COR: 1
                                                    TN: 1
```

2. If the **Meet-me Conferencing** field is set to **y** on the first page of this screen, go to the next page of the Vector Directory Number screen.

Figure 7: Vector Directory Number screen

```
change vdn 50300                                     Page 2 of 3
                                                    VECTOR DIRECTORY NUMBER
                                                    MEET-ME CONFERENCE PARAMETERS:
                                                    Conference Access Code: 123456
                                                    Conference Controller:
                                                    Conference Type: expanded
                                                    Route-to Number: 850280
```

3. In the **Conference Type** field, you can enter the conference type to be a **6-party** or **Expanded**. The default for this field is **6-party**.
4. In the **Route-to Number** field, enter the administered unique UDP extension for the VDN.

Note:

The Route-to Number can be created by adding an AAR routing code (e.g. 555) with a uniform-dialplan extension (e.g. 22xx), (where 22xx is designated as Net type aar.) For ease of tracking, the extension can be set to match the vdn number. In this case, vdn number 2201 can have a route-to number of 555-2201.

5. Save your changes.

Note:

The Route-to Number must be **unique** and can have a maximum of 16 digits. The Route-to Number must be unique for each VDN and be unique across all Expanded Meet-me Conference VDNs.

Expanded Meet-me Conferencing display updates

The Expanded Meet-me Conferencing feature provides real-time display updates for up to 30 conferees. Display updates for up to 30 is expected to meet the needs of most business conferences.

If Expanded Meet-me Conferencing size exceeds 30 parties, all other parties that join the conference see a fixed display that shows “CONFERENCE >30” in the **nnn** field of the end-user telephone. The display message can also be “>30 party conference in progress.”

Users can select the **conf-dsp** button to get details regarding number of parties on the conference, and scroll through the list of parties.

- If a user presses the **conf-dsp** button once, the user can view the conference count with the text **<nnn>-party conference** field.
- If a user presses the **conf-dsp** button again, the system provides updated information as other parties join the conference.
- Pressing the **conf-dsp** button again enables the user to scroll through the list of parties on a conference.

Note:

This applies to ad-hoc, six-party Meet-me Conferences and Expanded Meet-me Conferencing.

Note:

The **conf-dsp** button is located on the **Station** screen in the Button Assignments area. You can select this button from the drop down list.

Language translations are available for system displays. You can also administer this in a user-defined language by using the **display-messages-transfer-conference** command in the **Language Translations** screen.

End users can activate or deactivate certain system features and capabilities. End users can also modify or customize some aspects of the administration of certain features and capabilities. This section includes the following end-user procedures for Expanded Meet-me Conferencing:

- **Conf-dsp button** - The conf-display button shows real-time updates about an Expanded Meet-me Conference. For more information, see Expanded Meet-me Conferencing display updates on page 749 of the *Communication Manager Feature Description*.
- **Fe-mute** - Far end mute allows the user to selectively mute a party on an Expanded Meet-me Conference. If the conf-dsp button is administered, the user sees a fe-mute button on the display information and can use the far end mute capability.
- **Selective drop** - If the conf-dsp button is administered, a user can use the Drop button on the telephone or station to drop a party from a call. If you are an attendant, you can press the “forced-rel” button. This is an important capability in situations where a conference gets interrupted by a conferee putting the call on hold, or if the Music on Hold feature is active.

Appendix B: Post Installation Configuration Changes

All configuration is accomplished during installation. The following sections describe the configuration files that support the Meeting Exchange.

General System Configuration

The Meeting Exchange's default system information is located in a configuration file is named system.cfg. This file is stored in /usr/ipcb/config.

Follow this procedure to configure Meeting Exchange:

1. Open the system.cfg file for editing.
2. Locate the General system information section.
3. Edit the address and extension settings for the site. [Table 1](#) describes the settings.

Table 1: General System Information Settings

Setting	Description
IPAddress	The IP address of the Meeting Exchange server
MyListener	The SIP URI of the port and transport that Meeting Exchange listens on. To listen to a different port number, to the end of the entry add "####", where #### is the port number. To use a different transport, add "sips" for a secured transport or ";transport=tls" to the end of the entry.
respContact	This setting overwrites the default contact header on responses. If not set it sends the default Contact SIP header.
DiffServTOSValue	Value that is inserted into the IP header's type of service (TOS) field. This is used to differentiate the service. Some routers will route specific values faster or with priority.
EthernetVlanValue	Value that is inserted into the IP header's virtual LAN (VLAN) field.
sessionRefreshTimer	This setting tell the system how often to refresh sessions, guaranteeing that calls disconnect when the far end drops. The recommended setting is 180 seconds (3 minutes).
1 of 2	

Table 1: General System Information Settings (continued)

Setting	Description
minSetTimerValue	The minimum setting allowed for the sessionRefreshTimer. The system forces calls received with a lower refresh timer to use this value. This value should always be less than or equal to the sessionRefreshTimer setting. The recommended setting is 180 seconds (3 minutes).
MaxMeetingCount	Maximum meeting count.
MaxChannelCount	Maximum channel count.
2 of 2	

This example shows a system.cfg file:

```
# ip address of the server
IPAddress=10.221.10.192

# request we will be listening to
MyListener=sip:6000@10.221.10.192

# if this setting is populated,
# it will Overwrite the contact field in responses
respContact=
MaxChannelCount=300

# diff serv this value will appear on the TOS field of the IP packet
DiffServTOSValue=5
# vlan value
EthernetVlanValue=0
#refresh timer settings, to refresh the sessions that are connected
#recommended value is to set both at 180(3 minutes refresh)
sessionRefreshTimerValue=180
minSETimerValue=180
# Not used for UnixWare version of Meeting Exchange
#MaxMeetingCount=100
#MaxOperatorCount=10
```

4. Save the changes.

Software Media Server Configuration

The media server features are configured in the `softMediaServer.cfg`. This file is stored in `/usr/ipcb/config` directory.

This sample file contains field explanations:

```
#
# Media server runtime parameters
# Automatic gain on or off 1 or 0
# 0 turns AutomaticGain OFF;
# 1 turns AutomaticGain ON (default)

AutomaticGain=0

# generate comfort noise on =1 off = 0
# 0 turns ComfortNoiseGeneration OFF;
# 1 turns ComfortNoiseGeneration ON (default)

ComfortNoiseGeneration=0

# initial port for the rtp data, default is 42000
baseRtpPort=42000
```

Proxy Configuration

The proxy table contains the information on each proxy that Meeting Exchange registers with. Each line of the file contains the following settings:

Table 2: Proxy Configuration Settings

Setting	Description
ProxyURI	The valid SIP URI of the Proxy Server.
Contact	The default Contact SIP header used on dial outs or on the proxy registration.
1 of 2	

Table 2: Proxy Configuration Settings (continued)

Setting	Description
To	The default To SIP header used by the system when dialing out or on the proxy registration.
From	The default From SIP header used by the system when dialing out or on the proxy registration.
usrName	The user name of the person listed as the contact.
passWord	The password of the user.
refreshTime	Refresh time.
2 of 2	

Software Process Configuration

The processTable.cfg file defines all software processes started by ipcbinit. The file may contain comment lines, which start with “#.” Any blank lines in the file are ignored.

[Table 3](#) describes the fields contained in the processTable.cfg file.

Table 3: processTable.cfg

Field	Description												
processName Note: The extra "c" in processname.	<p>This is a string identifying which process to run. The process name starts an executable file, where:</p> <table border="1"> <thead> <tr> <th>Process Name</th> <th>Starts this Executable</th> </tr> </thead> <tbody> <tr> <td>bridget700</td> <td>bridgeTranslator</td> </tr> <tr> <td>msDispatcher</td> <td>msdispatcher</td> </tr> <tr> <td>sipagent</td> <td>sipagent</td> </tr> <tr> <td>mediaServer</td> <td>softms for the S6200, or convMs for Convedia.</td> </tr> <tr> <td>initipcb</td> <td>initipcb</td> </tr> </tbody> </table>	Process Name	Starts this Executable	bridget700	bridgeTranslator	msDispatcher	msdispatcher	sipagent	sipagent	mediaServer	softms for the S6200, or convMs for Convedia.	initipcb	initipcb
Process Name	Starts this Executable												
bridget700	bridgeTranslator												
msDispatcher	msdispatcher												
sipagent	sipagent												
mediaServer	softms for the S6200, or convMs for Convedia.												
initipcb	initipcb												
IpckeyNumber	A unique number used to identify the System V message queue used as the input for each process. To list all the IPC queues and the keys they use, enter the "ipcs -q" command.												
ProcessExe	The path to the executable file for this process. It is used by the initipcb process to automatically start and stop processes. If you do not want a process started or stopped then use "noexecute". Use noexecute for processes that are started by dcbinit such as bridget700.												
IpAddress	The IP address of the computer that runs this process. Enter the address in numeric form such as 10.221.10.192 or simply use "local" (without quotes) for processes running on the computer where the processTable.cfg file is located.												
Route	The destination processes for messages sent by the process being defined. When there are multiple routes, use a comma separated list.												
ProcessArgs	Command line arguments passed to a process. The use is specific to the process so there is no general guideline on what values to use.												

After installation, copy the appropriate version of processTable.cfg to /usr/ipcb/config/processTable.cfg. Usually, the field installer adds or deletes extra media servers. Other settings remain as installed.

Post Installation Configuration Changes

The process definition file will contain settings similar to these:

```
#process definition file. It enumerates the number of processes
# in the network.
# Contains the name of the process Key ID and the IP address.
bridget700      100   noexecute                local
  dspEvents/msDispatcher, netEvents/sipAgent
sipAgent        101   /usr/dcb/bin/sipagent     local
  dspEvents/msDispatcher, appEvents/bridget700
mediaServer    103   /usr/dcb/bin/softms       local
  dspEvents/msDispatcher, netEvents/msDispatcher1
msDispatcher   102   /usr/dcb/bin/msdispatcher local
  netEvents/sipAgent, appEvents/bridget700,dspEvents/mediaServer
initipcb       110   noexecute
```

Appendix C: System Log Messages

Describes messages generated by Meeting Exchange.

Meeting Exchange Errors

The following sections describe errors which are unique to the Replace variable w/ short product name server.

Debug Process Errors

Message: DEBUGPROCESS Couldn't find interface for %s
Meaning: Can not find information for named process in processTable.cfg.
Corrective Action: Either load backup copy of processTable.cfg or manually add the information for the missing process.

SIP Utility Messages

Message: Can't print a message of %d chars
Meaning: The software can handle user input messages that are 2-5 bytes, but not other lengths.
Corrective Action: No action required. The user should try to enter the expected number of digits.

Message: Media Server is not in the process table
Meaning: The media server process was not found in processTable.cfg
Corrective Action: Either load a known good backup copy of processTable.cfg or manually edit it to add a media server process.

System Log Messages

- Message: Error Couldn't find appsendto interface
Meaning: Could not find the process to which the media server sends messages.
Corrective Action: Either load a known good backup copy of processTable.cfg or manually edit it to add the to which the media server sends messages
- Message: Error Couldn't find DSP send to interface
Meaning: Could not find the process which handles network events.
Corrective Action: Either load a known good backup copy of processTable.cfg or manually edit it to add the process
- Message: Error Couldn't find the init process Key
Meaning: Could not find the IPC message queue ID in processTable.cfg
Corrective Action: Either load a known good backup copy of processTable.cfg or manually edit it to add the process
- Message: Error Creating IPC on start up : %s, strerror(errno)
Meaning: Failed to create an IPC message queue. The message contains the error number which is defined in /usr/include/sys/errno.h
Corrective Action: Make sure the software is running as root. If you logged in as another user to manually stop or start the software, you must first "su root". If that is not the problem, the system may be out of resources. The safest way to fix that is to reboot.
- Message: ERROR: msgctl() failed. errno=%d. qid=%d\n, errno, id
Meaning: Failed to increase the size of an IPC message queue
Corrective Action: Make sure the software is running as root. If you logged in as another user to manually stop or start the software, you must first "su root". If that is not the problem, the system may be out of resources. The safest way to fix that is to reboot.

Message: ERROR: msgctl() failed. errno=%d. qid=%d\n, errno, id
Meaning: Failed to read the size of an IPC message queue.
Corrective Action: Make sure the software is running as root. If you logged in as another user to manually stop or start the software, you must first "su root". If that is not the problem, the system may be out of resources. The safest way to fix that is to reboot.

Message: ERROR: msgsnd() failed. errno=%d. destKey= 0x%x q=%d total send errors %d\n,
Meaning: Failed to send an IPC message.
Corrective Action: Log in and use the "ipcs -qa" command.
Look for the message queue with the matching key.
Determine if the queue is full. If it is, then restart that process or the server software.
If the message queue doesn't exist, then restart the process which reads messages from that queue.
The system may be out of messages.

Message: [%d] Unknown event queue to process[%d].
Meaning: An IPC message contains an unknown message queue ID
Corrective Action: Check the configuration in processTable.cfg. Make sure the event queue and process are listed. Then check the process is running using the "dcbps" command line utility. If that looks good, then restart the software.

Message: No entry for the process %s ",interface.c_str()
Meaning: There was no record in processTable.cfg for the named process
Corrective Action: Either load a known good backup copy of processTable.cfg or manually edit it to add the information for the missing process.

INIT Messages

- Message: %s is not in the process table ,IPCBINIT_PROC_NAME
Meaning: There was no record in processTable.cfg for the named process.
Corrective Action: Either load a known good backup copy of processTable.cfg or manually edit it to add the information for the missing process.
- Message: started process=<%s> with pid=<%d>. Args=<%s/%s>
Meaning: This is only an error if preceded by a message saying a process ended (killed, aborted, exited, etc
Corrective Action: If a process is frequently restarting, it could be caused by: 1. Lack of some resource, 2. A timeout, 3. Internal error. Restart the software
- Message: unable to start process=<%s>
Meaning: The executable file for the process may not exist, or it may not be set to be executable, or you may have manually started the software as a normal user when you should start the software as "root".
Corrective Action: 1. Look for the name of the executable program in processTable.cfg.
2. Go to /usr/dcb/bin to see if the executable file exists.
3. See if the file is owned by "root" and the permission for the file includes an "x" for the file owner.
4. See if the system is out of virtual memory using the SCO administration utilities. If it is out of memory, then reboot
- Message: process <%s> not responding after %d tries
Meaning: A process is not replying to poll messages, probably because it is overloaded, or deadlocked.
Corrective Action: The software should automatically restart the process to recover. If not,
1. The system may be overloaded. Use the "ipcs -qa" message to see if the message queue has many messages in it. If that is the case, a faster CPU and more RAM may help.
2. There could be an internal error in that process.

Message: select() error=%d
Meaning: There was an error in the "select" command which is used to delay a few seconds. The error message includes an error code which is defined in /usr/include/sys/errno.h
Corrective Action: Look up the error code in /usr/include/sys/errno.h. Then take the corrective action implied by the error code.

Message: sleeping for another %x seconds
Meaning: The process is waiting for the specified period. This is not an error.
Corrective Action: None

Message: unknown queue id responded %d, recvdQueueID
Meaning: The program got a message with an unknown message queue ID.
Corrective Action: This may be caused by an internal software error.

Message: restarting process=<%s>,
Meaning: A process was automatically restarted. That is done to recover when a process ends unexpectedly, or when the init process kills a process that appears to be frozen.
Corrective Action: 1. The system may be overloaded. Use the "ipcs -qa" message to see if the message queue has many messages in it. If that is the case, a faster CPU and more RAM may help.
2. There could be an internal error in that process.

System Log Messages

Message:	Killed process=<%s> with pid=<%d>
Meaning:	A process was killed. Either because: 1. The process failed to respond to poll messages which implies it is either overloaded or stuck. 2. The system is shutting down.
Corrective Action:	Look for previous messages to see the sequence of events. If there are previous messages about this process not responding, then that process could either be overloaded or stuck. If the process repeatedly dies and restarts try restarting the software. If that fails to resolve the problem, try rebooting the machine. If that fails to resolve the problem,
Message:	Can't kill missing process=<%s> with pid=<%d>
Meaning:	The init process tried to kill a process that doesn't exist. Usually that means the process aborted or exited already
Corrective Action:	Normally no action is needed. The init process should automatically restart the process.
Message:	Error %d when killing process=<%s> with pid=<%d>
Meaning:	A system error occurred when killing a process.
Corrective Action:	Look up the error code in /usr/include/sys/errno.h. Take the corrective action implied by the error code.
Message:	Error %d in waitpid, errno
Meaning:	A system error occurred when trying to determine why a process ended.
Corrective Action:	Look up the error code in /usr/include/sys/errno.h. Then take the corrective action implied by the error code.
Message:	Process %s id %d exited normally with exit status %d
Meaning:	A program exited with the specified code. Normally a zero means the program ended normally while a nonzero number means an error occurred.
Corrective Action:	This message is only an error if a program ends when it is not expected to end. This message is normal when the system is shutting down.

Message: Process %s id %d killed by signal %d

Meaning: A program was killed by init or some other process, or by a signal generated by the process which died. The signal which killed the process is listed.

Corrective Action: Look up the signal in /usr/include/sys/signal.h. If the signal is SIGABRT, SIGBUS, or SIGSEGV that is an internal Spectel error so contact Spectel Support. The SIGTERM or SIGKILL signals are used by the init process. To see if those are related to a problem you need to look at the sequence of messages related to this process. If there are messages about a process not responding, then use the recommended corrective action for that.

Message: Process %s id %d stopped by signal %d

Meaning: A program was stopped by the specified signal. Normally a program is stopped by sending it a suspend signal. That is normally done from a shell and is not expected when the software is running in the background.

Corrective Action: Look up the signal in /usr/include/sys/signal.h. If the signal is SIGABRT, SIGBUS, or SIGSEGV that is an internal system error. The SIGTERM or SIGKILL signals are used by the init process. To see if those are related to a problem you need to look at the sequence of messages related to this process. If there are messages about a process not responding, then use the recommended corrective action for that.

Message: invalid message=%x

Meaning: The message type in the message is wrong.

Corrective Action: This is probably caused by either an internal error or an incompatible mixture of different processes.

System Log Messages

Appendix D: Troubleshooting

This section describes common problems and suggested resolutions.

General Issues

This section discusses general problems and also problems that are common to the operating system.

Obtaining Debug Information

The log files provide a key source of information. Additionally, you can use many Linux commands for troubleshooting.

[Table 4](#) lists commands that can assist you with resolving problems.

Table 4: Common Linux Commands

Type this command:	To:
<code>df -k</code>	determine if the system is running low on disk space.
<code>ifconfig -a</code>	list Ethernet interfaces.
<code>hostname</code>	find the computer's name
<code>ipcs -qa</code>	find the number of IPC messages used by each queue. The number in the "QNUM" column represents the number of messages waiting in a queue.
<code>kill</code>	stop a process.
<code>man -k <keyword></code>	search for help on a topic, using a given keyword.
<code>netstat -s</code>	get detailed statistics, including dropped packets, for all protocols; IP, UDP, ICMP, etc.
<code>ping <ip address></code>	test connectivity to another system on the network.
<code>ps -eaf</code>	list running processes.
1 of 2	

Table 4: Common Linux Commands (continued)

Type this command:	To:
<code>sar 5 5</code>	list available CPU.
<code>uname -a</code>	determine which kernel is running
<code>uptime</code>	identify the length of time a computer has been running
<code>who</code>	determine which users are logged onto the computer.
<code>who -b</code>	view the last reboot, date, and time.
2 of 2	

[Table 5](#) lists log files that can be helpful in diagnosing common problems.

Table 5: Log Files

Directory	File	File description
<code>/usr2/ipcb/log</code>	<code>various</code>	The daily log and user transaction log files.
<code>/var/log</code>	<code>messages</code>	The Linux system log.
<code>/var/log/</code>	<code>dmesg</code>	The Linux boot up log.

Restart a Program

Normally let the server software automatically start and stop its down processes. However if you need to manually restart a process, you will need to manually stop the process first.

Use the `kill` command to stop a process. This command requires the process ID number. To obtain the process ID number, use the `ps -eaf` command to list processes and locate the one you want to stop.

To start the process either let the `init` process automatically restart it or for debugging purposes you can type the command at the command line to start the process.

 **Tip:**

Manually starting a process is not recommended because there may be required command line parameters. Therefore manually starting a process should be done in consultation with the development staff.

Unable to create new files

Systems that are unable to create a new files may exhibit the problem in a number of ways. Typically the system may not be able to perform these functions:

- Write new log messages
- Edit a file

Administrators can use the `df` command with the `k` switch to determine if the system is running low on disk space.

```
df -k
```



Free up space by removing old log or core files from a file system that is more than 80% full.

System Configuration Problems

This section describes problems related to installation or configurations.

System does not work with Dual SMP CPUs

The software is only designed and tested for single CPU systems.

Slow System Performance

When the system's response time is slow, take these actions:

1. Check the amount of free CPU time using this command:

```
sar 5 5
```

Idle time should be 50% or more. If the idle time is less than 50%, list all running processes by typing:

```
ps -eaf
```

Note:

The idle percentage is an unreliable way of determining the system load. A better indicator is to verify that DTMF digits are handled promptly by making a test call to the system.

2. Look for the processes which use the most cumulative CPU.

**Tip:**

The system may be trying to handle more calls than is optimal when the softms process is using the most CPU time.

System does not Accept VoIP Calls

Follow this procedure:

1. Telnet to the system to make sure it is on the network and running.
2. List current processes:

```
ps -eaf
```
3. Verify that the listed processes include:
 - eventHandler
 - sipagent
 - appropriate media server process(es):
softms for software based DSP
4. Check the log files stored in /usr2/ipcb/log for reasons why the system is not accepting the calls. Some clues might include processes which are repeatedly stopping and restarting or SIP stack errors.
5. Change directories to usr/ipcb/config.

```
cd /usr/ipcb/config
```
6. Verify that processTable.cfg defines all processes that should be running for this configuration.
7. Verify system.cfg is correctly configured. See [General System Configuration](#) on page 31

VoIP programs restarted?

You can search the log file for the date you suspect the program restarted. For example, if you suspect that the VoIP programs restarted on July 28, follow these steps.

1. `. cd /usr2/ipcb/log`
2. `grep start system.log.Jul28`
3. Look for messages in the log which say "INIT started process..." and the date and time that occurred.

Network Problems

This section describes some common network-related problems.

Debug a Network Problem

Use these guidelines to determine the cause of a problem on the network.

1. Ensure the software media server can communicate with the network by logging on to a bridge and using the "ping" command to test communication with another computer on the network. If you get an error saying "Network is unreachable" refer [Network is unreachable](#) on page 50 before continuing.
2. Verify the IP address of both computers which are experiencing communication problems. Use the "ifconfig -a" command. Ignore the "lo" loop back device" and look for "inet" parameter for the "eth0" device.
3. Use "netstat" while trying to communicate with the system. If communication is working, the system should display a line describing a socket from the source computer to the destination (bridge computer). If successful, the state of the socket is "ESTABLISHED".

Note:

This step applies to TCP/IP communication not UDP.

If you are debugging VOIP configuration then refer to [System does not Accept VoIP Calls](#) on page 48.

Troubleshooting

4. View `/var/log/messages` file to see if there are any errors related to network communication problems.

If all the configuration files and parameters look favorable, then there may be a message format difference between the sending and receiving computers.

You will need to debug that you need some type of packet sniffer program or computer. See the tip for the symptom "How can I capture or monitor IP traffic?"

If you haven't located the problem gather all the information, and send it to the developers for further investigation.

Network is unreachable

Verify network connectivity by following this procedure:

1. List all Ethernet interfaces, by typing this command:

```
ipconfig -a
```

This command should return more than "lo" loop back. When it returns only the loop back, or displays the message "Network is unreachable", then install the driver for your Ethernet interface.

When the command returns:"eth0", "net0", or another network interface that has the state "DOWN", refer to "Need to configure Network interface"

- a. Confirm that the Ethernet cable is firmly connected to the back of the server and to the router/switch/hub.
- b. Verify the port on the router/switch/hub is enabled and correctly configured.

When the command returns:"eth0", "net0", or another network interface that has the state If the state is "Up":

- c. Verify the IP address of the interface is correct.
 - d. Test communication with other computers in the network by using the ping command. For example:

```
ping 10.221.10.254
```
2. If the problem remains, then reboot
 3. If the problem remains after rebooting, contact support.

Network is unreachable: Firewall

When calls are not connecting, verify that you are able to get through the firewall.

Meeting Exchange expects to open these inbound ports:

tls port 5061

Meeting Exchange expects to open these output ports:

udp ports 42000 through 42599

How can I capture or monitor IP traffic?

Use a second system to monitor IP traffic. The system should run Ethereal or tcpdump. Connect this system to an mirrored port on the Ethernet switch/router/hub. That port should be a mirror of all traffic sent to the VoIP system.

Audio Problems

This section provides procedures for trouble shooting common audio problems.

No Audio

When a call appears to connect, but no audio is heard, follow this procedure:

1. Check the SIP phone configuration used by the person who reported the problem.
2. Verify the correct media server program is running on the server:
 - softms for the media server
3. Check the system configuration file, system.cfg.

Debug Process

The debug process is named "ipcbdbg". This utility is stored in /usr2/ipcb/bin/ipcbdbg. Run it from the command line:

```
ipcbdbg all t # Turn on trace level debugging for all processes
```

Error Messages

This section provides guidelines for handling error messages.

SIP Utility Errors

Message: Can't print a message of %d chars.

This message is used when a conference user enters a series of digits that are to be spoken back to the user. There is a limit of 2-5 digits. If there are 6 or more digits then this error message should be displayed.

Test Procedure

1. Use a call script that requires entering digits that will be repeated to the user.
2. Enter 6 or more digits.
3. Look in the error log file.

Message: Media Server is not in the process table

Test Procedure

1. Kill initipcb if it is running.
2. Edit the processTable.cfg to change the mediaServer line to a comment by inserting “#” at the beginning of the line.
3. Start the initipcb process while logged in as “root”.

Message: DEBUGPROCESS Couldn't find appendto interface

1. Stop the Meeting Xchange software if it is already running.
2. Make a backup copy of processTable.cfg in the config directory.
3. Edit processTable.cfg to delete the list of interfaces after a process such as the line for softms.
4. See if the Meeting Xchange software is already running using “ps -eaf | grep initipcb”. If that process is running then kill it.
5. Start the Meeting Exchange software. On Linux you would run initipcb from the command line while logged in as “root”.
6. Check the error logs to find the message we are testing

Message: Error Couldn't find DSP send to interface

1. Kill initipcb if it is already running.
2. Start with a working copy of processTable.cfg.
3. Edit processTable.cfg to delete "eventHandler" from the list of interfaces on the line for the softms.
4. Start initipcb.
5. Check the error log.

Message: Error Couldn't find the init process Key

1. Kill initipcb if it is already running.
2. Start with a working copy of processTable.cfg.
3. Edit processTable.cfg to change the "initipcb" line to a comment.
4. Start initipcb.
5. Check the error log.

Message: Error Creating IPC on start up: %s, strerror(errno)

1. Kill initipcb if it is already running.
2. Write a C program that creates as many message queues as possible until it gets an error. Run that program on the Linux server for Meeting Xchange
3. Start initipcb.
4. Check the error log.

Message: Error: msgctl() failed. Errno=%d. quid=%d

You will normally never see this error message. To see it you need to exhaust the number of available messages in the system. A detailed test procedure has not yet been defined.

Message: Error: msgctl() failed. Errno=%d. quid=%d

Failed to read the size of a message queue. You may be able to create this error by:

1. Start MeetingXChange if it is not already running.
2. Wait about 30 seconds.
3. Delete a message queue using "ipcrm -q" and the appropriate message queue key.
4. Place a VOIP call.
5. Check the error message log.

Message: Error: msgsnd() failed. Errno=%d destKey=0x%x q=%d total send errors %d

1. Stop the Meeting Xchange software if it is running.
2. Write a C program to send numerous R_U_ALIVE message to a process such as eventHandler. Have the program keep sending as fast as it can forever.
3. Place a VOIP call.
4. Check the error log.

Message: [%d] Unknown event queue to process[%d]

1. Write a C program that sends an R_U_ALIVE message with an invalid source message queue ID.
2. Use that program to send a message to a process such as the SIP agent
3. Check the error log

Message: No entry for the process %s

1. Kill initipcb if it is already running.
2. Start with a working copy of processTable.cfg.
3. Edit processTable.cfg to change the “mediaServer” line to a comment.
4. Start initipcb.
5. Check the error log.

Message ipcbinit is not in the procesTable

1. Kill initipcb if it is already running.
2. Start with a working copy of processTable.cfg.
3. Edit processTable.cfg to change the “initipcb” line to a comment.
4. Start initipcb.
5. Check the error log.

Message: started process=<%s> with pid=<%d>,. Args=<%s/%s>

Most of this time this is not an error, rather it is information to let you know that a process started. To create this message:

1. Kill a process such as eventHandler
2. Initipcb should automatically restart the process in about 33 seconds.
3. Look at the error log.

Message: unable to start process=<%s>

While Meeting Xchange is running:

1. Rename an executable file such as eventhandler by doing: `mv /usr/ipcb/bin/eventhandler /usr/ipcb/bin/eventhandler.backup`
2. Kill the eventhandler process.
3. Wait over 33 seconds for the process to be restarted.
4. Look in the error log file.

Message: process <%s> not responding at %d tries

Same procedure as for “Message: unable to start process”

Message: select() error=%d

Normally you will never see this error.

Message: sleeping another %x seconds

This is not an error. It is informing the user the init process is waiting again before it polls running processes. This message is shown when the init process is interrupted while waiting on a select statement. That can happen if a signal is sent to initpcb.

1. While meeting Xchange is running
2. While logged in as “root” enter “kill -1” followed by the process number for initpcb
3. Repeat step 2 several times, waiting a few seconds in between repetitions
4. Check the error log

Message: unknown queue id responded %d

1. Write a C Program to send a response to an R_U_ALIVE message using an invalid message queue ID
2. While Meeting Xchange is running, use the test program to send the bad message
3. Check the logs

Message: restarting process

1. Kill a process such as eventHandler.
2. Initpcb should automatically restart the process in about 33 seconds.
3. Look at the error log.

Message: Killed process=<%s> with pid=<%d>

This is normally an information message and not an error message. This is displayed when initpcb kills a process. It does that in two cases:

1. When Meeting Xchange is stopping
2. When a process fails to respond to an R-U-Alive message

Note:

To see this error message stop Meeting Xchange and look at the log file.

Message: Can't kill missing process=<%s> with pid = <%d>

1. Start Meeting Xchange
2. In another window enter: `cd /usr/ipcb/bin`
3. Kill the eventhandler
4. In the other window start eventhandler using the debugger: `gdb eventhandler`
5. Set a break point at main using "b main"
6. Enter "Run"
7. At this point the eventhandler process is created by halted by the breakpoint.
8. Initpcb will try sending R_U_ALIVE messages which will not get a response for the eventhandler. To see this use "tail -f" to watch the log file.
9. Wait about 10 seconds after the first "not responding" message then exit the debugger to end the eventhandler process.
10. Later when ipcbinit tries to kill the eventhandler, that process is already gone so you will see the expected error.

Message: Error %d when killing process

This normally doesn't occur. Test procedure is TBD.

Message: Error %d in waitpid

This normally doesn't occur. Test procedure is TBD.

Message: Process %s id %d exited normally with exit status %d

1. Write a C program to perform only these commands:

```
main()
{
  exit(1);
}
```

2. Stop Meeting Xchange if it is running

3. Replace a real process such as eventhandler with the test process by renaming the executable to eventhandler while saving the original eventhandler executable.
4. Start Meeting Xchange.
5. Look for the message in the error log

Message: Process %s kd %d kill by signal

1. While Meeting Xchange is running, kill a process such as the eventhandler using “kill -15”
2. Look at the error log

Message: Process %s kd %d kill by signal

1. While Meeting Xchange is running, kill a process such as the eventhandler using “kill -19”
2. Look at the error log

Message: invalid message=%x

1. Write a C program to send message with an illegal message type such as -1. Have that program get the message queue ID from a command line parameter.
2. Use the test program to send the bad message to each process in the system.
3. Look at the error log.

Appendix E: Utilities

Describes command line utilities that support the EMMC.

Introduction

The server software includes utilities designed for configuring system features and troubleshooting system problems. All system utilities referenced in this appendix are located in the `/usr/ipcb/bin` directory.

Diagnostic Data for Meeting Exchange

Meeting Exchange includes a command line interface the “`ipinfo`” command that may be used to print information about the bridge, conferences, and lines.

Using ipinfo

To use the “`ipinfo`” command;

1. Open a terminal window on the bridge either by:
 - a. using telnet to login to the bridge. Or alternately,
 - b. using the system console to log in and open a terminal window.

Note:

The `ipinfo` command must be run with root privileges so that it will have the proper access permissions to update the log files.

2. Type the `ipinfo` command using the syntax described in the following section.

Note:

Once you enter the `ipinfo` command, the system normally pauses for five seconds to gather all responses from the event handler. Do not stop the process (by using Control-C). If the process is stopped for any reason, refer to [Lock file](#) on page 61 for information on clearing a locked file.

In general there is one line of output for each line, participant, meeting, etc. This done makes it easy to use `grep`, `sed`, `awk`, and other text processing utilities to search for values.

Most of the displayed information is dynamically updated. However, configuration information is static. Configuration information can be reviewed by viewing the configuration files stored in `/usr/ipcb/config`.

Syntax

This section describes the `ipinfo` syntax.

```
ipinfo [-b | -c | -h | -l | -m | -md | -ml | mld | -p] [ ID list |
DNIS string ]
```

where:

[ID list] = [number | numeric range | comma separated list | or a combination of those three]

The ID numbers generally start at 1 and increment to the maximum number supported by the Event Handler. If the ID list is:

- provided, the default lists information for all objects used by a meeting or participant.
- not provided, the default normally lists information for all active objects—those which are in use by a meeting or participant.

[DNIS string] = a string of digits used to dial into a conference call.

The following table describes the parameters.

Table 6: ipinfo Parameters

Parameter	Description / Example
	When <code>ipinfo</code> is used without a parameter, it displays the help information.
<code>-b</code>	Display properties for the bridge. This command does not require a list of IDs. Example: <code>ipinfo -b</code>
<code>-c</code>	Display properties for the active and inactive conferences that are listed in the list of line IDs. When you do not provide a list of conference IDs, all possible conferences display.
<code>-h</code>	Display help information that lists these parameters and descriptions.
<code>-l</code>	Display properties for the lines that are listed in the list of line IDs.
1 of 2	

Table 6: ipinfo Parameters (continued)

Parameter	Description / Example
-m	Display properties for the meetings that are listed in the list of meeting IDs. Tip: Use “ipinfo -m” to display information for all active meetings. Example: ipinfo -m 1-3,5,50-99
-md	Display conference properties given a single DNIS string.
-ml	Display properties for all participants in the meetings listed in the list of meeting IDs.
-mld	Display the list of participants in a meeting given a DNIS string.
-p	Display properties for the participant(s) listed in the participant ID list. Example: ipinfo -p 100
[ID List]	[number numeric range comma separated list or combination of those three]
2 of 2	

Lock file

Only one person at a time can use the “ipinfo” command. To enforce that, a temporary lock file is created call “/tmp/ipinfoLock”.

If someone kills the “ipinfo” command before it is finished the lock file may remain. In that case it must be manually deleted before the “ipinfo” command.

Configuration Files

Normally the configuration files provided with the release and never modified by users. However, if a customer wants to customize the output of a command, that can be accomplished by editing the associated default property file.

Utilities

The ipinfo command gets the default list of properties from the file listed in [Table 7](#). Those files are stored in the /usr3/ipcb/config directory on Linux.

Table 7: Property files

Object type	Used by	Default property file
Bridge	ipinfo -b	bridge.props
Conference	ipinfo -c	conference.props
Line	ipinfo -l	line.props
Meeting	ipinfo -m	meeting.props
Participant	ipinfo -p	participant.props

Bridge Properties

Sample output from “ipinfo -b”:

```
ActiveMeetingCount:0 MaxMeetingNum:100 MaxOperatorNum:10  
MaxParticipantNum:300
```

Conference Properties

Sample output from “ipinfo -c”:

```
ConfNum:0 ConferenceID:1 IsRecordPaused:0 IsRecording:0  
ConfNum:0 ConferenceID:2 IsRecordPaused:0 IsRecording:0
```

(Similar data for ID 3-97 not shown)

```
ConfNum:0 ConferenceID:98 IsRecordPaused:0 IsRecording:0  
ConfNum:0 ConferenceID:99 IsRecordPaused:0 IsRecording:0
```

Meeting Properties

Example from “ipinfo -m” or “ipinfo”:

```
Active:1 AutoHangup:1 Duration:0 InstantMeetingId:0 Lecture:0  
MaxDuration:0 MaxMeetingSeconds:0 MeetingID:1 MeetingSeconds:0  
ModeratorCount:0 ModeratorList:( ) Music:64 Secure:0 SecureAllow:1 UserCount:0  
UserList:( 1 ) WaitingForModerator:0
```

In this example, there are no moderators so the list of moderator Ids is empty “()”, and there is one user with an ID of 1.

Meeting List Properties

Example from “ipinfo -ml”. This lists all members for the specified meeting ID(s). If no meeting ID list is provided then by default it lists data for all participants for all meetings.

```
Active:0 Agc:0 CallContact:<sip:10.221.10.176:5060>
CallFrom:Spectel.com <sip:Spectel.com>;tag=167640536
CallRequestUri:sip:1234*1212#@10.221.10.189 CallTo:<sip:1234*1212#@10.221.10.189>
ChannelID:0 Codec:pcmu/8000
Conference:0 DialInProgress:0 LineID:1 LocalAudioPort:12004
MeetingID:1 Music:0 Mute:0 Protocol:RTP/AVP RemoteAudioPort:8000 Sdp_o:- 572892160
572892160 IN IP4 10.221.10.18 UserType:User
```

Line Properties

This “ipinfo -l” example only displays the active lines.

```
Active:0 Agc:0 CallContact:<sip:10.221.10.176:5060>
CallTo:<sip:1234*1212#@10.221.10.189> ChannelID:0 Conference:0 DialInProgress:0
LineID:1 Music:0 Mute:0
```

Participant Properties

This example, “ipinfo -p”, displays the active participants:

```
CallContact:<sip:10.221.10.176:5060> CallTo:<sip:1234*1212#@10.221.10.189> ConfMusic:0
ConfMute:0 LineMusic:0 LineMute:0 ParticipantID:1 SelfMute:0 isInMeetingConf:0
```


Index

Symbols

/usr/dcb	48
/usr/ipcb/bin	59
/usr/ipcb/config	31
/usr2/ipcb	46 , 49
/usr2/ipcb/bin/ipcbdbg	51
/var/adm	46

C

Commands

Linux

df	45
fconfig	45
hostname	45
ipcs	45
kill	45
man	45
netstat	45
ping	45
ps	45
sar	46
uname	46
uptime	46
who	46

commands, *see* commands *under individual feature names*

Communication Manager

Configure Server	9
upgrade	19
Configuration	33
problems	47
proxy	33
Configure Server.	9

D

Debug.	45
df	45

E

EMMC

upgrade	19
-------------------	--------------------

Expanded Meet-me Conferencing

administering	
prerequisites	22
description	21

External DNS Server Configuration	13
---	--------------------

F

fconfig	45
Firewall	51

H

hostname	45
--------------------	--------------------

I

Individual IP Services	13
Individual Services	11
Install	15 , 17
log file	17
message log	18
Interfaces	11
ipcb_cdinst.sh.log	17
ipcb_cdinst.sh_verbose.log	17
ipcs	45
ipinfo	59

K

kill	45
----------------	--------------------

L

Linux

commands

df	45
fconfig.	45
hostname	45
ipcs	45
kill.	45
man	45
netstat.	45
ping	45
ps	45
sar	46
uname.	46
uptime.	46
who	46
Log Files.	46

Index

M

man.	45
Meet-me Conference	
commands	
display system-parameters customer-options .	22
screens	
Optional Features	22
messages	18
Modem Interface.	14

N

netstat	45
Network Time Server.	14

P

ping.	45
processTable.cfg	34 , 35
Proxy	
configure.	33
ps.	45

S

sar.	46
Signaling Group	23
Software Media Server	33
configure	33
Switches.	12
System	
log messages	37
system.cfg	31

T

Trunk	23
-----------------	--------------------

U

uname	46
Upgrade	19
Communication Manager	19
EMMC	19
uptime	46

W

who	46
---------------	--------------------