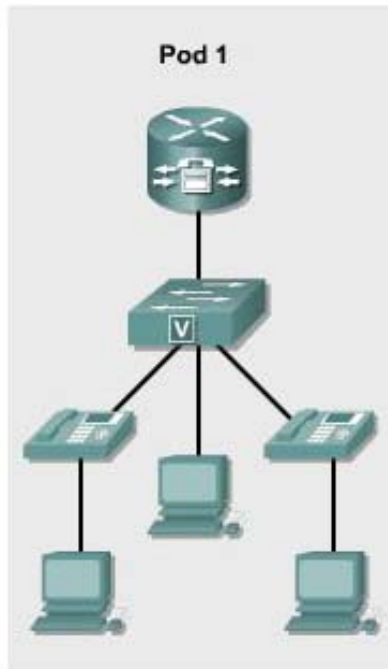


## Lab 3.1.2 CME Manual Phone Setup



### Objective

- Configure IP Phones using the manual configuration process

### Equipment Requirements

- Cisco CallManager Express (CME) capable router with specific files for IP phone (basic CME .tar file)
- Inline power capable switch or non-inline power switch with power injectors
- Workstation with an Ethernet 10/100 NIC installed
- One Cisco IP phone

In this lab the ACME.com Company has decided to use the manual setup process to configure the Cisco CallManager Express router and phones. Use IOS commands to achieve the following goals:

- Manually configure the IP phones with an extension and connect them to the network
- Assign a name to the IP phones

This lab relies on labs 2.1.1, 2.1.2, and 2.1.3 being successfully completed. The lab uses information referenced in IP Telephony Table 1 and IP Telephony Table 2.

## Step 1 Verify if the Telephony Service is Running

- If necessary, put the basic configuration from 3.1.1 on the router and switch.
- Ensure that **NO** phones are connected to the switch at this time.
- Access the Cisco CallManager Express router and use the **show running-config | begin tele** command to verify that the telephony service has not been configured. The **begin tele** parameter will search the running configuration for the first instance of the letter combination of tele, which is really meant to be telephony-service. If no telephony-service is configured nothing will be displayed and the CMERouterX prompt will be seen again. If a configuration exists, use the **no telephony-service** command to erase any existing configuration.

```
CMERouterX# show running-config | begin tele
CMERouterX(config)# no telephony-service
```

## Step 2 Configure DHCP for the IP Phones

- Enter **ip dhcp exclude-address 10.X5.0.1 10.X5.0.10** (where **X** is the pod number) to disallow certain addresses from a DHCP pool that is about to be created in the next step.

```
CMERouterX(config)# ip dhcp excluded-address 10.X5.0.1 10.X5.0.10
```

- Enter the command **ip dhcp pool CMEVoiceX** (where **X** is the pod number) to name the DHCP pool and begin configuration of the parameters that will be served to remote devices.

```
CMERouterX(config)# ip dhcp pool CMEVoiceX
```

- Use the **network 10.X5.0.0 255.255.255.0** command to set up the range of addresses that will be used.

```
CMERouterX(dhcp-config)# network 10.X5.0.0 255.255.255.0
```

- Enter the **default-router 10.X5.0.1** (where **X** is the pod number) to assign a default gateway to the IP phone.

```
CMERouterX(dhcp-config)# default-router 10.X5.0.1
```

- With CallManager Express, the IP phones receive their initial configuration information and phone firmware from the TFTP server associated with the CME router. The phones usually get the IP address of their TFTP server using the DHCP **option 150** command. When using CME, the TFTP server address obtained by the phone should point to the CME router IP address.

Enter **option 150 ip 10.X5.0.1** to assign the TFTP server address as the CME router. The Cisco IP phone downloads the image configuration files from this address (the router). The configuration files will be located on the voice-capable router.

```
CMERouterX(dhcp-config)# option 150 ip 10.X5.0.1
```

## Step 3 Verify and Configure Firmware

- From privileged mode, use the **show flash:** command to verify the IP phone firmware files that are present. The firmware files for a 7960 and 7940 will start with "P003." For example, P00303020214.bin. Do not continue to Step 4 until the IP phone firmware files are present in flash.

**For a 28x1 router:** If these files are not there, there is a file that the instructor must obtain from the Cisco site that starts with the letters "cme-basic" and has a file extension of ".tar". An example of the filename is cme-basic-123-11T.tar.

**Note:** The .tar file must match the IOS version on the CME router.

The file must be extracted and uploaded to the router Flash memory from a TFTP server. Copy the .tar file into the appropriate TFTP server folder. From privileged mode, use the **archive tar /xtract tftp://tftp\_server\_ip\_address/.tar\_filename flash:**. An example of this command is as follows: **archive tar /xtract tftp://10.3.0.33/cme-basic-123-11T.tar flash:**.

**For a 1760 router (on an older IOS):** If these files are not there, the instructor must download the IOS zip file and the Pxxx files are part of the zip file. For this model of router and this IOS, upload any and all files that start with P003 from the downloaded file to the router Flash memory using a TFTP server. An example of this command is **copy tftp:P00303020214.bin flash:**.

- b. In the space provided, write down the firmware files present in flash memory that relate to IP phones (the P003 files).

```
CMERouterX# show flash:
```

---

---

#### Step 4 Basic Phone Configuration

- a. From global configuration mode, use the **tftp-server flash:P003XXXX.XXX** command to allow the firmware files to be accessed through the TFTP server service running on the router. An example of this command is as follows: **tftp-server flash:P00303020214.bin**

```
CMERouterX(config)# tftp-server flash:P003XXXXXXXXX.XXX
```

**Note:** The filename typed after the **flash:** parameter is case-sensitive.

Repeat this command for all "P003" firmware files present in Flash memory.

- b. From global configuration mode, enter telephony service mode.

```
CMERouterX(config)# telephony-service
```

- c. To see the maximum allowable number of ephones (another name for IP phones) that this system can use enter the command **max-ephones ?**.
- d. In the space provided, write the maximum number of phones supported by the current router:

```
CMERouterX(config-telephony)# max-ephones ?
```

---

- e. Set the maximum number of IP phones to two, as this will be sufficient for the classroom lab.

```
CMERouterX(config-telephony)# max-ephones 2
```

The **max-dn** command specifies the maximum number of directory numbers or virtual voice ports supported by the router. The maximum number is IOS and platform-dependent. The default number is zero.

To see the maximum allowable number of directory numbers that this system can use enter the command **max-dn ?**.

```
CMERouterX(config-telephony)# max-dn ?
```

- f. In the space provided, write the maximum number of directory numbers supported by the router and IOS being used:

```
CMERouterX(config-telephony)# max-dn ?
```

---

- g. Use the **max-dn** command to configure the maximum number of directory numbers to 20, as this will be sufficient for the classroom lab.

```
CMERouterX(config-telephony)# max-dn 20
```

- h. The **load** command updates the Cisco IOS Telephony Services (Cisco ITS) configuration file for a specific type of IP phone to add the name of the correct firmware file that the phone should load. This filename also provides the version number for the phone firmware that is in the file. Cisco IP phones update themselves with new phone firmware whenever they are initially started or reloaded. When a phone is started or rebooted, the phone reads the configuration file to determine the name of the firmware file it should load and then looks for that firmware file on a TFTP server.

**Note:** A separate **load** command is needed for each type of phone (although the 7940 and 7960 phones have the same firmware and share the **7960-7940** keyword).

Another keyword used with this command is the name of the firmware file. With the **load** command, do not specify the file extension.

Specify the firmware version and associate it with a model of phone by entering the command **load model firmware\_filename\_without\_extension**. An example of this command when using a 7940 and/or 7960 IP phone model is **load 7960-7940 P00303020214**.

**Note:** The tar filename used with the **load** command is entered *without* the filename extension (such as .bin). Also, the filename is not case-sensitive with this command. Before using the **load** command, view the model numbers available by typing **load ?**.

```
CMERouterX(config-telephony)# load ?
```

```
CMERouterX(config-telephony)# load model filename
```

Was the 7960-7940 model number used as part of the command? If not, why not? \_\_\_\_\_

---

- i. The IP phone will attempt to transfer a configuration file that it needs called XmlDefault.cnf.xml. This file is automatically generated by the CME router through the use of the **ip source-address** command and is placed in router memory. The XmlDefault.cnf.xml file contains the IP address that the phones used to register for service. The phones use the SCCP (Skinny Client Control Protocol) to do this. SCCP is commonly referred to as simply Skinny or the Skinny protocol.

Enter the **ip source-address** command to define the address and port number where the Cisco CallManager Express router is listening for registrations (Skinny messages). This address should correspond to a valid CME router IP address. Note the **X** in the command is the port number.

```
CMERouterX(config-telephony)# ip source-address 10.X5.0.1 port 2000
```

- j. Use the **create cnf-files** command to build XML configuration files that will be used by the phones during the boot process.

```
CMERouterX(config-telephony)# create cnf-files
```

- k. What message(s) did the router display when the **create cnf-files** command was entered?

---

---

- l. The **keepalive** command sets the time interval (in seconds) between messages that are sent from the phone to the CME router. The default is 30 seconds, which is normally adequate. If the default is set to too large a value, it is possible that notification will be delayed when a system goes down.
- m. What is the maximum number of seconds that can be set using the **keepalive** command and how long is this in hours?

---

- n. Set the keepalive interval to 10 seconds.

```
CMERouterX(config-telephony)# keepalive 10
```

- o. Use the **show running-config | begin tele** command to view the results of the manual configuration.

```
CMERouterX# show running-config | begin tele
```

- p. Ephone-dn stands for Ethernet phone directory number. It represents a line that connects a voice channel to a phone so that calls can be made. An ephone-dn has one or more extensions or phone numbers associated with it to allow calls to be made. An ephone-dn is similar to a phone line. Each ephone has a number to identify it during the configuration process.

The **ephone-dn** command creates one virtual voice port and one or more dial peers. This command automates the process of associating dial peers. The number of ephone-dns that are created corresponds to the number of simultaneous calls that can be made because each ephone-dn represents one virtual voice port in the router. This means that if you want more than one call to the same number to be answered simultaneously, multiple ephone-dns (virtual voice ports) are needed.

There are two types of ephone-dns: single-line and dual-line. On the most basic level, the single-line option has one phone number associated to it. A dual-line can have one or two numbers associated with it. The single-line option should be used when phone buttons have a one-to-one correspondence to the regular phone lines that come into a CME system. The dual-line option allows making two call connections at the same time using one phone line button.

Ensure no IP phones are plugged into the switch.

From global configuration command, type **ephone-dn 1 ?**

- q. What two options are available? \_\_\_\_\_
- r. Add an ephone-dn for the first line appearance on the first phone in the pod by entering the **ephone-dn 1 dual-line** command. The **dual-line** parameter defines the type of ephone-dn is being created.

```
CMERouterX(config)# ephone-dn 1 dual-line
```

In ephone-dn mode enter the **number X000** command (where X is the pod number).

```
CMERouterX(config-ephone-dn)# number X000
```

- s. Enter a name that will be associated with this DN by entering the **name *firstname lastname*** command. Either make up a name or use a student's name. Example **name Ray Hampton**.

```
CMERouterX(config-ephone-dn)# name firstname lastname
```

Exit e-phone-dn configuration mode by typing **exit**.

- t. From global configuration mode, enter the command **ephone 1** to enter ephone configuration mode for the first phone in the pod.

```
CMERouterX(config)# ephone 1
```

- u. The MAC address of the IP phone will be needed in the next step. The MAC address is on a sticker on the bottom of the phone. The sticker is normally located in the center. Upon close inspection, the letters MAC are stenciled directly to the left of the white sticker that contains the MAC address. In the space provided, write down the MAC of the first IP phone.



- v. Assign the MAC address to ephone 1 with the **mac-address H.H.H** (where H is the four hexadecimal characters). Note that the periods must be inserted between each set of four characters, and the letters are not case-sensitive. An example entry is **mac-address 0013.c43b.4999**.

```
CMERouterX(config-ephone)# mac-address H.H.H
```

- w. The **button** command is used to define properties to the buttons located to the right of the IP phone's LCD. The **button** command has a number that follows it with the number 1 representing the top button on the IP phone. The number is followed by a separator character that specifies phone characteristics. For example, the colon separator assigns the phone a normal ring, a single pulse for internal calls and a double pulse for external calls.

```
CMERouterX(config-ephone)# button 1:1
```

## Step 5 Verify Phone Registration

- a. From privileged mode, enter the **debug ephone register** command.

```
CMERouterX# debug ephone register
```

- b. Plug in the IP phone with MAC address configured previously.
- c. View the ephone registration debugging output. This may take a few moments. Verify that the phone has registered and the proper DN appears with the line.
- d. Once the phone has registered, what is the last line of the **debug** output?

- 
- e. Enter **undebug all** to turn off all debugging.

```
CMERouterX(config)# undebug all
```

- f. Save the router configuration. It will be used to complete the next lab.