



APPENDIX **B**

Using Cisco IPICS for External Notifications

The Cisco policy engine notification action, which is described in [Chapter 6, “Using the Cisco IPICS Policy Engine,”](#) contacts designated recipients and provides them with information that you specify. Those recipients must be configured in Cisco IPICS.

You can also use Cisco IPICS to send notifications to recipients who are not configured in Cisco IPICS. This type of notification is called an *external notification* and does the following:

1. Simultaneously calls many external users at telephone numbers that Cisco IPICS obtains from a file that you specify.
2. Plays a designated message to each user who answers the call.
3. Captures results of each call in a log file that you can review at any time.

You invoke an external notification by sending an HTTP request or by posting a Common Alerting Protocol (CAP) .xml file to the appropriate server.

This appendix describes how to set up Cisco IPICS for external notifications and how to invoke external notifications. It includes these sections:

- [Requirements for Implementing External Notifications, page B-1](#)
- [Invoking External Notifications, page B-3](#)
- [Retrieving External Notification Results, page B-5](#)
- [Reviewing External Notification Results, page B-8](#)
- [XML File Parameters for External Notifications, page B-8](#)

Requirements for Implementing External Notifications

To use Cisco IPICS for external notifications, you must configure a dial engine, designate a list of recipients, and designate a message file to play to the recipients.

The following sections describe these requirements in detail.

Dial Engine

A dial engine is a Cisco IPICS server the system to which you send the request to invoke an external notification and that handles dialing out and playing messages to external notification recipients. This server enables distribution of the call load so that notifications to a large number of recipients can be made efficiently. Any Cisco IPICS server in a network can be a dial engine.

To configure a Cisco IPICS server as a dial engine, you edit a .xml configuration file on that server and set the dialEngine sub-element attributes for the dial engine. To do so, perform the following procedure:~~

Procedure

-
- Step 1** On the Cisco IPICS server that you want to be the dial engine, edit the dedicated-dial-engines.xml file, which is stored in this folder:
- /opt/cisco/ipics/tomcat/current/webapps/ipics_server/WEB-INF/classes
- Step 2** In the dedicated-dial-engines.xml file, uncomment the “dialEngine” sub-element, and set the following attributes for the dial engines:
- id—Set a unique identifier for a dial server (the identifier can be the host name of the server)
 - bulkNotifyApp—Set to BulkNotifyDialer
 - host—IP address or host name of the dial engine
- Step 3** Save the dedicated-dial-engines.xml file.
- Step 4** Restart the Cisco IPICS service by entering the following command from the root user ID:
- service ipics restart.**
-

Recipient List

A recipient list is a .xml file that contains a list of each person who should receive the external notification message. To invoke an external notification that contacts these recipients, you must know the URL of the server on which the recipient list resides.

A recipient list should be in the format that is shown in this section and should follow these guidelines:

- “id” is required and must be unique for each recipient.
- “name” is optional.
- “contacts” is a list of recipients in the order that they should be called.
- “phone type” is optional.
- “dn” is required. The dn value is sent to the voice gateway exactly as entered.

The following is an example of a valid recipient list:

```
<users>
  <user id="5" name="Kim Smith">
    <contacts>
      <phone type="mobile" dn="14085554444" />
      <phone type="home" dn="14085555555" />
    </contacts>
  </user>
  <user id="7" name="Bob Jo">
    <contacts>
      <phone type="work" dn="1342433534" />
      <phone type="home" dn="14085543255" />
    </contacts>
  </user>
</users>
```

Message File

A message file is a .wav file in pulse code modulation (PCM) or CCITT u-Law format that contains the recorded message to play to recipients. Cisco recommends that the message be no longer than 90 seconds.

To invoke an external notification that plays this message, you must know the URL of the server on which the message file resides.

Invoking External Notifications

You can invoke a notification action by sending an HTTP get request to the dial engine or by posting a valid CAP .xml file over HTTP to the dial engine.

Invoking an External Notification by Sending an HTTP Request

To invoke an external notification with an HTTP request, send the following request to the server that is designated as the dial engine:

```
http://<Dial_Engine_Server>:8080/ipics_server/BulkNotify?
method=httpNotify?capId=<CAP_ID_#>&userListURL=<User_List_URL>
&waveFileURL=<Wave_File_URL>
&announcementText=<Announcement_Text>
&questionWaveURL<number>=<Question_Wave_URL>
```

where:

- *<Dial_Engine_Server>* is the fully qualified hostname or the IP address of the Cisco IPICS server that includes the dial engine.
- *<CAP_ID_#>* is a unique identifier that you assign to this invocation request.
- *<User_List_URL>* is the HTTP URL from which the recipient list can be retrieved.
- *<Wave_File_URL>* is the HTTP URL from which the message file can be retrieved.
- (Optional) *<Announcement_Text>* is the text that displays on Cisco Unified IP Phones.
- *<number>* is the question number. This value must start with 1 and increment by values of one.
- (Optional) *<Question_Wave_URL>* is the HTTP URL from which the question file can be retrieved. The retrieved question file name must conform to the following format:

Question<number>_<dtmf_choices>.wav

where:

- *<number>* is the question number. This number must match the number in the questionWaveURL attribute.
- *<dtmf_choices>* is the number of allowable keypad digits that the phone recipients can enter, as shown in the following questionWaveURL parameter examples:
 - An HTTP request with the parameter &questionWaveURL1=Question1_1234.wav plays audio to which the phone recipient can respond by pressing 1, 2, 3, or 4 on their keypad.
 - An HTTP request with the parameter &questionWaveURL2=Question2_12.wav plays audio to which the phone recipient can respond by pressing 1 or 2.

- An HTTP request that contains the parameter `&questionWaveURL3=Question3_1.wav` plays audio to which the phone recipient can only respond by pressing 1. You can use this type of request to confirm that a user has received this notification.

**Note**

Parameter that include spaces or special characters must use standard HTTP escape codes.

Invoking an External Notification by Posting a CAP XML File

You can invoke an external notification by posting a valid CAP .xml file to this URL:

**http://<Dial_Engine_Server>:8080/ipics_server/BulkNotify?
method=capNotify?&cap=<CAP_XML>**

where:

- `<Dial_Engine_Server>` is the host name or IP address of the Cisco IPICS server that includes the dial engine.
- `<CAP_XML>` is a valid CAP file

To use this notification method, the following guidelines must be followed:

- A resource with the resource description `IPICSUserList` must exist. The Uniform Resource Identifier (URI) for this resource must specify the HTTP URL of the server from which the recipient list can be retrieved. The mimeType for this resource must be `text/xml`.
- A resource with the resource description `IPICSWaveFile` must exist. The URI for this resource must specify the HTTP URL of the server from which the message file can be retrieved. The mimeType for this resource must be `audio/wav`.
- (Optional) A resource with the resource description `AnnouncementText` may exist. The URI for this resource must specify the HTTP URL of the server from which the text message can be retrieved. The mimeType for this resource must be `text/plain`.
- (Optional) A resource with the resource description `QuestionWaveURL<number>` may exist. The URI for this resource must specify the HTTP URL of the server from which question file can be retrieved. Please see HTTP invocation section for further info on how to specify Question files. The mimeType for this resource must be `audio/wav`.

The following is an example of a valid CAP file:

```
<?xml version="1.0" encoding="UTF-8"?>
<alert xmlns="urn:oasis:names:tc:emergency:cap:1.1">
<identifier>5</identifier>
<sender>mycontact@mycompany.com</sender>
<sent>2003-06-11T22:39:00-07:00</sent>
<status>Actual</status>
<msgType>Alert</msgType>
<source>SW</source>
<scope>Public</scope>
<info> <category>Rescue</category>
<event>Earthquake</event>
<urgency>Immediate</urgency>
<severity>Severe</severity>
<certainty>Likely</certainty>
<eventCode>
<valueName>SAME</valueName>
<value>CAE</value>
</eventCode>
<senderName>San Jose Earthquake Prevention</senderName>
```

```

<headline>SJ Earthquake</headline>
<description>Earthquake in SJ Downtown</description>
<contact>Earthquake Prevention - 408-555-1212</contact>
<resource>
<resourceDesc>IPICSUserList</resourceDesc>
<uri>http://ipics-server:8080/BulkCapSimulator/BulkCapSimulator?method=userList</uri>
</resource>
<resource>
<resourceDesc>IPICSWaveFile</resourceDesc>
<uri>http://ipics-server:8080/BulkCapSimulator/BulkCapSimulator?method=waveFile&p=announcement</uri>
<mimeType>audio/wav</mimeType>
</resource>
<resource>
<resourceDesc>QuestionWaveURL1</resourceDesc>
<uri>http://ipics-server:8080/BulkCapSimulator/BulkCapSimulator?method=waveFile&p=q1</uri>
<mimeType>audio/wav</mimeType>
</resource>
<resource>
<resourceDesc>AnnouncementText</resourceDesc>
<uri>http://ipics-server:8080/BulkCapSimulator/BulkCapSimulator?method=announcementText</uri>
<mimeType>text/plain</mimeType>
</resource>
<area>
<areaDesc>San Jose, Ca</areaDesc>
<geocode>
<valueName>SAME</valueName>
<value>006037</value>
</geocode>
</area>
</info>
</alert>

```

Retrieving External Notification Results

After you invoke an external notification, you can retrieve a file that displays the results of that notification. To obtain the results of an external notification, send the following request to the Cisco IPICS server that includes the dial engine functionality:

http://<Dial_Engine_Server>:8080/ipics_server/BulkNotify? method=incidentStatus&capId=<CAP_ID_#> &statusType=<SUMMARY | DETAILED>

where:

- <Dial_Engine_Server> is the host name or IP address of the Cisco IPICS server that includes the dial engine.
- <CAP_ID_#> is the unique identifier number of the invocation request for which you are requesting results
- SUMMARY or DETAILED is the level of the status information that you want to receive.

A summary type of SUMMARY provides basic information about the incident including the start time, completion time, and status of the notification. A summary type of DETAILED provides a list of the users that were successfully reached, and how they responded to the questions that you provided.

The following is an example of output from a DETAILED-level status .xml file:

```

<?xml version="1.0" encoding="UTF-8"?>
<NotificationStatus>
<Incident capId="5">
<Status>SUCCESSFULLY_FINISHED</Status>
<StartTime>Mon Nov 19 15:25:07 PST 2007</StartTime>
<FinishTime>Mon Nov 19 15:25:17 PST 2007</FinishTime>
<UserSummary failedToNotify="0" successfullyNotified="1" totalUsers="1">
<User id="1028">
<Contacts>
<Phone dialNumber="1029" extn="" shared="false">
<Status>SUCCESSFUL</Status>
<Time>Tue Dec 04 16:35:55 PST 2007</Time>
<QuestionResponses>
<Response promptId="q1">
<Value>1</Value>
</Response>
</QuestionResponses>
</Phone>
</Contacts>
</User>
<User id="1028-3">
<Contacts>
<Phone dialNumber="1028" extn="" shared="true">
<Status>SUCCESSFUL</Status>
<Time>Tue Dec 04 16:35:50 PST 2007</Time>
<QuestionResponses>
<Response promptId="q1">
<Value>1</Value>
</Response>
</QuestionResponses>
</Phone>
</Contacts>
</User>
<User id="1028-1">
<Contacts>
<Phone dialNumber="1028" extn="" shared="true">
<Status>SUCCESSFUL</Status>
<Time>Tue Dec 04 16:35:50 PST 2007</Time>
<QuestionResponses>
<Response promptId="q1">
<Value>1</Value>
</Response>
</QuestionResponses>
</Phone>
</Contacts>
</User>
<!-- End detailed summary>
</UserSummary>
</Incident>
</NotificationStatus>

```

Table B-1 describes the summary report fields that display in the .xml file.

Table B-1 .xml File Summary Report Fields

Field	Description
Incident	This element contains the capId attribute. The capId attribute represents the unique identifier number of the invocation request for which you are requesting results.
Status	This element represents the status of the invocation request. Valid entries in this field are SUCCESSFULLY_FINISHED, PROCESSING, ABORTED_NO_CONFIGURED_DE, ABORTED_TIMEOUT_WAITING_ON_DE, ABORTED_FAILED_UPLOAD_PROMPT_TO_DE, ABORTED_PROMPT_DOWNLOADED_TRANSCODE_FAILURE, ABORTED_USER_LIST_PARSE_ERROR and ABORTED_UNKNOWN.
StartTime	This element represents the date and time that the invocation request started.
FinishTime	This element represents the date the time that the invocation request completed.
UserSummary	The UserSummary element contains the following attributes: <ul style="list-style-type: none"> failedToNotify provides you with the number of the participants that could not be notified. successfullyNotified provides you with the number of the participants that were successfully notified. totalUsers provides you with the total number of users in the notification.

If you specify a detailed report, the XML file contains the additional fields that [Table B-2](#) describes.

Table B-2 .xml File Summary Additional Report Fields for Detailed Report

Field	Description
User id	This element represents the ID of the phone number that was called. If more than one user shares a phone number, the shared attribute is set to true; otherwise it is set to false.
Phone	The Phone element contains the following attributes: <ul style="list-style-type: none"> The dialNumber field represents the phone number that was called. The extn field represents the extension of the number that was called. If the number does not contain an extension, this field is blank. The shared field indicates whether the number is shared by more than one user. If the shared field is set to true, the number is shared by multiple recipients; otherwise it is set to false.
Status	This element represents the status of the call. Valid fields are SUCCESSFUL, NO_ANSWER, BUSY, INVALID, NO_RESOURCE, ANSWERED_NO_ACKNOWLEDGEMENT and UNSUCCESSFUL.

Table B-2 *.xml File Summary Additional Report Fields for Detailed Report (continued)*

Field	Description
Time	The time that the end user device received the notification.
QuestionResponses	<p>The QuestionResponses element contains the following fields:</p> <ul style="list-style-type: none"> Response promptId is the ID of the question that you configured in the HTTP invocation request. This ID corresponds to the number in the name of the .wav file. For example, if the .wav file had the name Question1_<dtmf_choices>.wav, the response promptId is q1. Value is the number that the user pressed on the dial pad.

**Note**

If you do not specify the summaryType parameter, Cisco IPICS uses SUMMARY, as the default value.

Reviewing External Notification Results

Cisco IPICS captures and stores the results of calls that it makes when an external notification executes. This information is stored in the ipics.log file, which is stored on the Cisco IPICS server in this folder:

/opt/cisco/ipics/tomcat/current/logs

To stop/terminate an external notification with an HTTP request, send the following request to the server that is designated as the dial engine:

http://<Dial_Engine>:8080/ipics_server/BulkNotify?method=stopNotification&capId=<CAP_ID_#>

where

- <Dial_Engine> is the host name or IP address of the dial engine.
- <CAP_ID_#> is a unique identifier that you assign to this invocation request.

XML File Parameters for External Notifications

The dedicated-dial-engines.xml file includes the configurable parameters that [Table B-3](#) describes.

Cisco recommends that you do not change these values unless it is required by your network configuration. For example, if your network has high latency that causes calls to exceed a timeout value, you could increase the timeout value.

In addition, Cisco recommends that you do not modify any of the other values in this XML file, specifically the following files: maxDialEngineWaitForDialing, maxDialEngineWaitForAck, maxDialEngineWaitForIPPhone, and maxDialEngineWaitForOnCall.

Table B-3 *New Configurable Parameters in dedicated-dial-engines.xml File*

Value	Description
connectTimeout	<p>This value represents the maximum timeout value, in milliseconds, for the primary dial engine to connect to the dedicated dial engine. If the primary dial engine does not connect to the dedicated dial engine within the specified time, Cisco IPICS reports this error in the ipics.log file as a dial engine failure and requeues the notification request.</p> <p>To view the ipics.log file, navigate to the Serviceability > System Logs window in the Cisco IPICS Administration Console, and view the information that displays in the</p>
dialEngineFailureCapForDuration	<p>This value represents the time window, in milliseconds, that Cisco IPICS counts multiple dial engine failures as a single dial engine failure. If you use the default value of 500, Cisco IPICS counts any dial engine failures that occur during a half-second time window as one dial engine failure.</p>
initialEngineFailureReattemptDuration	<p>This value represents the time, in milliseconds, that Cisco IPICS waits before contacting the primary dial engine after a dial engine failure (for example, if there are no dial engine resources available to send the notification).</p> <p>Note Each subsequent retry attempt doubles this value. If you use the default value of 2000, and Cisco IPICS cannot complete a notification action because of a primary dial engine failure, Cisco IPICS waits 2 seconds before contacting the dial engine to retry the notification action. If Cisco IPICS detects another dial engine failure, it waits 4 seconds before contacting the dial engine.</p> <p>A successful dial engine call resets this value to the configured value.</p>
maxDialEngineFailureWaitTimeCap	<p>This value represents the maximum cumulative time, in milliseconds, that Cisco IPICS attempts to contact the dial engine. This value prevents the initialEngineFailureReattemptDuration value from incrementing to infinity, which can cause the dial engine to become unusable.</p> <p>If Cisco IPICS cannot contact the dial engine within the specified time, Cisco IPICS stops the external notification and reports the error in the ipics.log file.</p>
maxDialRetryCount	<p>This value represents the maximum number of times that Cisco IPICS attempts to resend a notification action to a user if the initial attempt does not succeed (for example, if a phone was busy or did not answer).</p>
maxWaitForAcquiringDialEngine	<p>This value represents the maximum time, in milliseconds, that the primary dial engine waits before stopping an external notification if no dedicated dial engines are available.</p>
phoneDialReadTimeout	<p>This value represents the maximum timeout value, in milliseconds, for the dedicated dial engine to respond to a user dial request from the primary dial engine. If the dedicated dial engine does not respond to the primary dial engine within the specified time, the notification action stops and Cisco IPICS reports the error in the ipics.log file.</p>

Table B-3 *New Configurable Parameters in dedicated-dial-engines.xml File (continued)*

Value	Description
promptUploadRead Timeout	This value represents the maximum timeout value, in milliseconds, for the dedicated dial engine to completely upload media (for example, a .wav file) from the primary dial engine. If the dedicated dial engine does not upload the media within the specified time, the notification action stops and Cisco IPICS reports the error in the ipics.log file.
redialPeriod	This value represents the time, in milliseconds, that Cisco IPICS waits to resend a notification action to a user if the initial attempt does not succeed.