Device Capability Query via CTI Feature

Feature Description

The Device Capability Query via CTI feature was added for Cisco Unified Communications Manager Release 8.0(1).

A backend CTI application that communicates with the phone using the UserData tunnel cannot retrieve information on device capabilities such as XSI feature support on a phone. Due to this lack of information, and to ensure compatibility, only a minimum set of features were generally configured.

The Device Capability Query via CTI feature overcomes this limitation. This feature allows a CTI-based application or a Cisco Unified Communications Manager application to query a registered phone for device capabilities using the UserData tunnel interface of the phone (over SCCP or SIP and RemoteCC).

Applications that have an HTTP interface with a phone do not have this limitation. The HTTP request from such phones include XSI capabilities header, and the DeviceInformationX servlet of such phones can be accessed to retrieve other device information.

Although designed to work using CTI over the UserData tunnel, this feature can also work over HTTP using the POST method.

Supported IP Phones and Codecs

The following table lists the Cisco Unified IP Phone models that support the Device Capability Query via CTI feature.

Table 1: Phone Models that Support the Device Capability Query via CTI Feature

Phone model	Supported, Not supported	Firmware supported (see note)
Cisco Desk Phone 9800 series **Note** Not supported on all Multiplatform phones		
9841	Supported	PhongOS 3.0(1) and later
9851	Supported	PhongOS 3.0(1) and later
Cisco IP Phone 8800 Series **Note** Not supported on all Multiplatform phones		
8811	Supported	10.2(2) and later
8841, 8851, 8861	Supported	10.2(1) and later
8851NR	Supported	10.3(1) and later
8845, 8865	Supported	10.3(2) and later
8865NR	Supported	11.7(1) and later
8875, 8875NR	Supported	PhoneOS 2.1 and later
Cisco IP Conference Phones **Note** Not supported on all Multiplatform phones		
8831	Supported	9.3(3) and later
8832	Supported	12.0(1) and later
7832	Supported	12.0(1) and later
Cisco Wireless IP Phone 8820 series		
8821	Supported	11.0(1) and later
Cisco IP Phone 7800 Series **Note** Not supported on all Multiplatform phones		
7811	Supported	10.3(1) and later
7821	Supported	9.1(1) and later
7841	Supported	9.1(1) and later
7861	Supported	9.1(1) and later

Note Cisco recommends the use of latest firmware. The firmware can be downloaded from the following location (requires login or service contract):

http://software.cisco.com/download/navigator.html?i=!mmd

Although several codecs are listed within the schema, only the codecs G711, G729, and G722 are currently supported.

XML Object Changes

To support this feature, new request and response objects are created. The `<getDeviceCaps>` is the request object and the `<getDeviceCapsResponse>` is the response object.

On receiving the `<getDeviceCaps>` object, the phone returns the `<getDeviceCapsResponse>` object. All elements in the `<getDeviceCapsResponse>` object are required and must not be null.

Schema Definition

The getDeviceCapsResponse XML schema is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.example.org/devicecaps"
xmlns:tns="http://www.example.org/devicecaps" xmlns="http://www.w3.org/2001/XMLSchema">
 <element name="getDeviceCapsResponse" type="tns:deviceCapType" nillable="true"/>
 <complexType name="deviceCapType">
  <all>
   <element name="physical" type="tns:physicalCapType" nillable="true"/>
   <element name="services" nillable="true">
    <complexType>
     <complexContent>
      <extension base="tns:servicesCapType">
       <attribute name="sdkVersion" type="string" use="required"/>
      </extension>
     </complexContent>
    </complexType>
   </element>
  </all>
 </complexType>
 <complexType name="physicalCapType">
  <all>
   <element name="modelNumber" nillable="false">
    <simpleType>
     <restriction base="string">
      <maxLength value="32"/>
      <minLength value="1"/>
     </restriction>
    </simpleType>
   </element>
   <element name="display" nillable="true">
    <complexType>
     <attribute name="width" type="unsignedShort" use="required"/>
     <attribute name="height" type="unsignedShort" use="required"/>
     <attribute name="bitDepth" type="unsignedShort" use="required"/>
     <attribute name="isColor" type="boolean" use="required"/>
    </complexType>
    </element>
   </all>
  </complexType>
  <complexType name="servicesCapType">
   <all>
    <element name="browser" type="tns:browserCapType" nillable="true"/>
   </all>
  </complexType>
  <complexType name="browserCapType">
   <all>
    <element name="accept" nillable="false"/>
    <element name="acceptLanguage" nillable="false"/>
    <element name="acceptCharset" nillable="false"/>
   </all>
  </complexType>
</schema>
```

Request and Response Examples for getDeviceCaps

The following are the request and response examples for a getDeviceCaps object:

Request sent to the phone:

```
<getDeviceCaps/>
```

Response returned from the phone:

```
<getDeviceCapsResponse>
  <physical>
    <modelNumber>CP-7970</modelNumber>
    <display width="298" height="168" bitDepth="12" isColor="true"/>
  </physical>
   <services sdkVersion="5.0.3">
    <browser>
 </services>
</getDeviceCapsResponse>
```

Troubleshooting

The following error may occur in this feature:

- If the getDeviceCaps object is invalid (misspelled), a parsing error is generated and a CiscoIPPhoneError object (with Number="1") is returned as the response.

Error Handling

Standard XML services debugging techniques are applied to this feature.

The root cause for any parsing errors is displayed in the phone console logs. For HTTP requests and responses, sniffer traces and web server debug can be used to examine the getDeviceCaps object to ensure that it conforms to the schema.