



Device Capability Query Via CTI

Feature Description

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

Applications that have a 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 via CTI over the UserData tunnel, this feature can also work over HTTP using the POST method.

Supported IP Phones and Codecs

[Table 1](#) lists the Cisco Unified IP Phone models that support the Device Capability Query via CTI feature:

Table legend:



- —Supported
- —Not supported

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




Phone Model	Supported/Not Supported	Firmware Supported ¹
9900 Series Phones		
9971		9.0(1) and later
9951		9.0(1) and later
8900 Series Phones		
8961		9.0(1) and later
7900 Series Phones		

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

Phone Model	Supported/Not Supported	Firmware Supported ¹
7905	✘	—
7906	✔	8.4(1) and later
7911	✔	8.4(1) and later
7912	✘	—
7931	✔	8.4(1) and later
7937	✘	—
7940	✘	—
7941	✔	8.4(1) and later
7942	✔	8.4(1) and later
7945	✔	8.4(1) and later
7960	✘	—
7961	✔	8.4(1) and later
7962	✔	8.4(1) and later
7965	✔	8.4(1) and later
7970	✔	8.4(1) and later
7971	✔	8.4(1) and later
7975	✔	8.4(1) and later
7985	✘	—
7900 Series Wireless Phones		
7920	✘	—
7921	✔	1.0(3) and later
7925	✔	1.3(1) and later
7925G-EX	✔	1.4 and later
7926	✔	1.4 and later
6900 Series Phones		
6921	✔	9.1(1) and later
6941	✔	9.1(1) and later
6945	✔	9.1(1) and later
6961	✔	9.1(1) and later
Other Devices		
Cisco IP Phone Communicator	✘	—

1. Cisco recommends the use of latest firmware. The firmware can be downloaded from the following location (requires login and/or service contract):
<http://tools.cisco.com/support/downloads/pub/Redirect.x?mdfid=278875240>.

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

Changes in XML Object

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>
          <complexContent>
            <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"/>
          </complexContent>
        </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>
```

```

    </all>
  </complexType>
</schema>

```

Example

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>
    <modelName>CP-7970</modelName>
    <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.