

# Getting System properties in MIDP

## Description

This document lists system properties that can be retrieved by using the `system.getProperty()` method call. The [File:SystemProperties.zip](#) related to this article can be used for checking the system properties that devices support.

If a system property string returns "null", the feature is not supported. Some of the system properties might return "null" even if the related API is supported. This is because the listed system properties include properties which are listed in the Mobile Service Architecture (JSR-248) specification. For example, the Series 40 3rd Edition, FP2 devices have SATSA-APDU package (JSR-177), but the "microedition.satsa.apdu.version" system property returns "null".

## CLDC 1.0 & 1.1, MIDP 1.0 & 2.0, and JTWI 1.0

String	Description
<code>microedition.profiles</code>	For MIDP 2.0 devices, this property MUST contain at least "MIDP-2.0".
<code>microedition.configuration</code>	The J2ME configuration supported by this device. For example, "CLDC-1.0".
<code>microedition.locale</code>	The name of the current locale on this device. For example, "en-US".
<code>microedition.platform</code>	The name of the host platform or device. In Nokia devices the name consists of "Nokia", the device model, and software version separated by "/". There is no space between "Nokia" and the model number nor on either side of "/". Formally, the syntax of the platform string is: <code>Nokia MODEL_NUMBER "/" SW_VERSION</code> . For example, "Nokia6310i/4.42" or "Nokia3510i/p1.25".
<code>microedition.encoding</code>	Returns the platform's default character encoding. In Nokia devices this value is "ISO-8859-1".
<code>microedition.comports</code>	A comma-separated list of ports is returned which can be combined with a comm: prefix as the URL string to be used to open a serial port connection.
<code>microedition.hostname</code>	Defined in MIDP 2.0 <code>javax.microedition.io.SocketConnection.getLocalAddress()</code> method. Returns the local hostname, if it is available.
<code>microedition.jtwi.version</code>	To identify a compliant device and the implemented version of this specification, the value of the system property <code>microedition.jtwi.version</code> must be "1.0".
<code>microedition.msa.version</code>	Version number of the supported MSA Specification, the possible values are "1.0" or "1.0-SUBSET" for implementations conforming to this specification.

## Package discovery

String	Description
<code>microedition.media.version</code>	If the optional package is supported, a version string is returned. Otherwise "null".
<code>microedition.pim.version</code>	If the optional package is supported, a version string is returned. Otherwise "null".
<code>microedition.m3g.version</code>	If the optional package is supported, a version string is returned. Otherwise "null".
<code>microedition.location.version</code>	If the optional package is supported, a version string is returned. Otherwise "null".
<code>bluetooth.api.version</code>	If the optional package is supported, a version string is returned. Otherwise "null".

<code>microedition.io.file.FileConnection.version</code>	If the optional package is supported, a version string is returned. Otherwise "null".
<code>microedition.global.version</code>	If the optional package is supported, a version string is returned. Otherwise "null".
<code>microedition.chapi.version</code>	If the optional package is supported, a version string is returned. Otherwise "null".
<code>microedition.sip.version</code>	If the optional package is supported, a version string is returned. Otherwise "null".
<code>wireless.messaging.version</code>	If the optional package Wireless Messaging API (JSR-120/205) is supported, a version string is returned. Otherwise "null".
<code>microedition.amms.version</code>	If the optional package Advanced Multimedia Supplements (JSR-234) is supported, a version string is returned. Otherwise "null".
<code>microedition.m2g.version</code>	If the optional package Scalable 2D Vector Graphics API for J2ME (JSR-226) is supported, a version string is returned. Otherwise "null".
<code>microedition.payment.version</code>	If the optional package Payment API (JSR-229) is supported, a version string is returned. Otherwise "null".
<code>microedition.contactless.version</code>	If the optional package Contactless Communication API (JSR-257) is supported, a version string is returned. Otherwise "null".
<code>microedition.sensor.version</code>	If the optional package Mobile Sensor API (JSR-256) is supported, a version string is returned. Otherwise "null".
<code>obex.api.version</code>	If the optional package Object Exchange Protocol (OBEX) API (JSR-82) is supported, a version string is returned. Otherwise "null".

## Mobile Media API (JSR-135)

String	Description
<code>supports.mixing</code>	A query for whether audio mixing is supported. The string returned is either "true" or "false". If mixing is supported, the following conditions are true: <ul style="list-style-type: none"> <li>At least two tones can be played with <code>Manager.playTone</code> simultaneously.</li> <li><code>Manager.playTone</code> can be used at the same time when at least one <code>Player</code> is playing back audio.</li> <li>At least two <code>Players</code> can be used to play back audio simultaneously.</li> </ul>
<code>supports.audio.capture</code>	A query for whether audio capture is supported. The string returned is either "true" or "false".
<code>supports.video.capture</code>	A query for whether video capture is supported. The string returned is either "true" or "false".
<code>supports.recording</code>	A query for whether recording is supported. The string returned is either "true" or "false".
<code>audio.encodings</code>	The string returned specifies the supported capture audio formats. Each format is specified in a special syntax. The formats are delimited by at least one space. For example, "{{{1}}}".
<code>video.encodings</code>	The string returned specifies the supported capture video formats. Each format is specified in a special syntax. The formats are delimited by at least one space. For example, "{{{1}}}".
<code>video.snapshot.encodings</code>	Supported video snapshot formats for the <code>getSnapshot</code> method in <code>VideoControl</code> . The string returned specifies the supported capture image formats. Each format is specified in a special syntax. The formats are delimited by at least one space. For example, "{{{1}}}".
<code>streamable.contents</code>	Returns formats that can be streamed (i.e., playing can be started before the whole content is loaded to the memory). Returns "null" if streamable formats do not exist. No formats support streaming at the moment.

## Wireless Messaging API (JSR-120, JSR-205)

String	Description
wireless.messaging.sms.smsc	The address of the SMS expressed using the syntax expressed by the msisdn item of the following BNF definition: {{{1}}}
wireless.messaging.mms.mmsc	Returns the address of the MMSC on devices with JSR-205 implemented.

## FileConnection API (JSR-75)

String	Description
fileconn.dir.photos	This points to the directory where photos captured with an integrated camera or other images are stored. For example, "file:///c:/My files/Images/".
fileconn.dir.videos	The same as the previous entry but with reference to videos. Downloaded videos are also stored here by default. For example, "file:///c:/My files/Video clips/".
fileconn.dir.tones	Ring tones and other similar audio files are stored in this directory. For example, "file:///c:/My files/Tones/".
fileconn.dir.memorycard	Root directory of a memory card in case it is available. For example, "file:///d:/".
fileconn.dir.private	Private work directory of MIDlet suite. For example, "file:///c:/System/MIDlets/[1015f294]/scratch".
fileconn.dir.photos.name	Localized UI name for the photos directory. For example, "Images".
fileconn.dir.videos.name	Localized UI name for the videos directory. For example, "video clips".
fileconn.dir.tones.name	Localized UI name for the sounds directory. For example, "Sound clips".
file.separator	File separator character. In Nokia devices: "/".
fileconn.dir.memorycard.name	Localized UI name for the memory card directory. For example, "Memory card".

## Java™ APIs for Bluetooth Wireless Technology (JSR-82)

**Note:** With the Bluetooth API, the `LocalDevice.getProperty()` method should be used to retrieve system properties; for example:  
`LocalDevice.getProperty("bluetooth.connected.devices.max")`

String	Description
bluetooth.l2cap.receiveMTU.max	This points to the directory where photos captured with an integrated camera or other images are stored.
bluetooth.connected.devices.max	The maximum number of connected devices supported (will include parked devices). The string will be in Base10 digits.
bluetooth.connected.inquiry	Is inquiry allowed during a connection? Valid values are "true" and "false".
bluetooth.connected.page	Is paging allowed during a connection? Valid values are "true" and "false".
bluetooth.connected.inquiry.scan	Is inquiry scanning allowed during a connection? Valid values are "true" and "false".
bluetooth.connected.page.scan	Is page scanning allowed during a connection? Valid values are "true" and "false".
bluetooth.master.switch	Is master/slave switch allowed? Valid values are "true" and "false".
bluetooth.sd.trans.max	Maximum number of concurrent service discovery transactions. The string will be in Base10 digits.
bluetooth.sd.attr.retrieveable.max	Maximum number of service attributes to be retrieved per service record. The string will be in Base10 digits.

## Scalable 2D Vector Graphics API (JSR-226)

String	Description
<code>microedition.m2g.svg.version</code>	The version of SVG that is supported by the underlying implementation. For example, if the underlying SVG engine conforms to the W3C SVGT 1.1 specification, the returned string is "1.1", and for SVGT 1.2 it is "1.2", and so on.
<code>microedition.m2g.svg.baseProfile</code>	The base profile of SVG that is supported by the underlying implementation. It is "tiny" for JSR-226 v1.0 implementations.

## Mobile 3D Graphics API for J2ME™ (JSR-184)

The Mobile 3D Graphics API properties can be queried with the `Graphics3D.getProperties()` method. It returns a `HashTable` of the properties and their values. Properties "maxViewportWidth" and "maxViewportHeight" were added in M3G 1.1, in case of M3G 1.0 this MIDlet returns "null" for these properties. Property "m3gRelease" is not mandated by the specification, but it is an implementation-specific property.

The example values are from an S60 3rd Edition device. See the JSR-184 specification for more information about the details of 3D graphics. The specification is available at [Nokia Developer - JSR-184](#).

String	Description
<code>supportAntialiasing</code>	Returns "true" if antialiasing is supported, otherwise "false". For example, "false".
<code>supportTrueColor</code>	Returns "true" if true color rendering is supported, otherwise "false". For example, "false".
<code>supportDithering</code>	Returns "true" if dithering is supported, otherwise "false". For example, "false".
<code>supportMipmapping</code>	Returns "true" if mipmapping is supported, otherwise "false". See <code>Texture2D</code> class description in JSR-184 specification for more information. For example, "true".
<code>supportPerspectiveCorrection</code>	Returns "true" if perspective correction is supported, otherwise "false". See <code>PolygonMode</code> class description in JSR-184 specification for more information. For example, "true".
<code>supportLocalCameraLighting</code>	Returns "true" if local camera lightning is supported, otherwise "false". See <code>PolygonMode</code> class description in JSR-184 specification for more information. For example, "false".
<code>maxLights</code>	Returns the maximum value of light sources. Minimum requirement for this is 8. See <code>Light</code> class description in JSR-184 specification for more information. For example, "8".
<code>maxViewportWidth</code>	Returns the maximum value of viewport width. Minimum requirement for this is 256. See <code>Graphics3D.setViewport()</code> method description in JSR-184 specification for more information. For example, "null".
<code>maxViewportHeight</code>	Returns the maximum value of viewport height. Minimum requirement for this is 256. See <code>Graphics3D.setViewport()</code> method description in JSR-184 specification for more information. For example, "null".
<code>maxViewportDimension</code>	Returns the minimum of { <code>maxViewportWidth</code> , <code>maxViewportHeight</code> }. Minimum requirement for this is 256. For example, "1024".
<code>maxTextureDimension</code>	Returns the maximum allowed size for a texture image. This is specific to each implementation. Minimum requirement for this is 256. See <code>Texture2D</code> class description in JSR-184 specification for more information. For example, "256".
<code>m3gRelease</code>	Returns the release date for the JSR-184 implementation. For example, "04_wk49".
<code>maxSpriteCropDimension</code>	Returns the maximum crop rectangle size. The minimum requirement for this is 256. See <code>Sprite3D</code> class description in JSR-184 specification for more information. For example, "256".
<code>numTextureUnits</code>	Returns the amount of texture units supported by the device. The minimum requirement for this is 1. See <code>Appearance</code> class description in JSR-184 specification for more information. For example, "2".

<code>maxTransformsPerVertex</code>	Returns an implementation defined limit to the number of bones (bone = a node in the skeleton group to transform the vertices with) that can have an effect on any single vertex. The minimum requirement for this is 2. See <code>SkinnedMesh</code> class description in JSR-184 specification for more information. For example, "4".
-------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Security and Trust Services API (SATSA) for Java™ 2 Platform, Micro Edition (JSR-177)

String	Description
<code>microedition.smartcardslots</code>	Indicates the names of the smart card slots. The value returned is a comma-separated list of the smart card slots which can be used in the <code>Connector.open()</code> string to identify the specific smart card slot. A typical configuration for a cold-swappable SIM card and a hot-swappable removable card would be 0C,1H.
<code>microedition.satsa.apdu.version</code>	Version of the SATSA-APDU API supported by the device. For example, "1.0".
<code>microedition.satsa.crypto.version</code>	Version of the SATSA-CRYPTO API supported by the device. For example, "1.0".
<code>microedition.satsa.pki.version</code>	Version of the SATSA-PKI API supported by the device. For example, "1.0".

## Location API for J2ME™ (JSR-179)

String	Description
<code>fileconn.dir.landmarks</code>	Returns the location of JSR-179 landmarks store in the file system.
<code>fileconn.dir.landmarks.name</code>	Returns localized name for the landmarks store.

## Advanced Multimedia Supplements API (JSR-234)

String	Description
<code>supports.mediacapabilities</code>	The String returned specifies the supported capabilities, delimited by one whitespace (Unicode U+0020). At least one of the possible values MUST be returned. The possible values are: music audio3d imageencoding imagepostprocessing camera tuner
<code>tuner.modulations</code>	The String returned specifies the supported tuner modulation settings for which a <code>Player</code> can be created. Returned modulations are delimited by one whitespace (Unicode U+0020). If no modulations are supported, the <code>tuner.modulations</code> returns "null". <code>TunerControl</code> specifies the following constants, but supported modulations are not limited to these: fm for FM audio radio am for AM audio radio
<code>audio.samplerates</code>	The String returned specifies the recommended audio sampling rates in Hertz, delimited by one whitespace (Unicode U+0020). Other sampling rates might be supported as well, but these are recommended. If the audio output is not supported, the <code>audio.samplerates</code> returns

	"null".
audio3d.simultaneouslocations	<p>The String returned specifies the maximum number of SoundSource3D modules that the application is recommended to have active simultaneously in a typical game-playing situation when graphics rendering (2D or 3D) and reverberation are also taking place. A SoundSource3D module is active when it contains one or more Players in the PREFETCHED or STARTED states. The implementation does not guarantee that this number of SoundSource3Ds may be active simultaneously under all conditions; it is just a hint to the application. (It may even be possible for more than this number of SoundSource3Ds to be active simultaneously, but this is not guaranteed either.)</p> <p>The number returned presumes that every active SoundSource3D has exactly one Player, and that the Player's content type is a monophonic linear WAV with a sample rate of 16 kHz and 16 bits per sample.</p>
camera.orientations	<p>The String returned specifies the orientations of the cameras of the device. The String consists of orientations per device, delimited by one whitespace (Unicode U+0020). The String for orientation has the format device_name: orientation without whitespace characters.</p> <p>An example of a returned String: "devcam0:outwards devcam1:inwards".</p> <p>The orientation values can be the following:</p> <ul style="list-style-type: none"> <li>inwards (the camera is pointing towards the user when the device is held in a natural way.)</li> <li>outwards (the camera is pointing away from the user when the device is held in a natural way.)</li> <li>unknown (the pointing direction of the camera is user configurable or cannot be known for some other reason.)</li> </ul> <p>If no cameras are supported, the camera.orientations returns "null". Default camera (capture://video) is listed first in the String.</p>
camera.resolutions	<p>The String returned specifies the sensor sizes of the cameras of the device. The String consists of sensor sizes per device, delimited by one whitespace (Unicode U+0020). The String for a sensor size has the format device_name : width x height without whitespace characters.</p> <p>An example of a returned String: "devcam0:1024x768 devcam1:640x480".</p> <p>If no cameras are supported, the camera.resolutions returns "null". Default camera (capture://video) is listed first in the String.</p>

## J2ME™ Web Services API (JSR-172)

String	Description
xml.jaxp.subset.version	Version of JAXP Subset APIs supported by the device, for example, "1.0".
xml.rpc.subset.version	Version of JAX-RPC Subset APIs supported by the device, for example, "1.0".

## S60 3rd Edition, FP2 system properties

String	Description
com.nokia.mid.imei	The IMEI (International Mobile Equipment Identity) of the device. See <a href="#">IMEI in Wikipedia</a> .
com.nokia.mid.imsi	The IMSI (International Mobile Subscriber Identity) stored in the SIM card. For more information on IMSI, see <a href="#">IMSI in Wikipedia</a> . Note that the access to this system property is limited with com.nokia.mid.mobinfo.IMSI permission. By default, this permission is available only in manufacturer and operator domain.

com.nokia.mid.networkid	Network identification parameters, such as network ID (NID in CDMA and MNC in GSM) and network short name. The returned string identifier contains both in the following format: <network ID> (<network short name>).
com.nokia.mid.networksignal	Current (GSM/CDMA) network signal strength. The output format is as follows: <the number of bars the phone should display>( <the signal strength in dBm>dBm).
com.nokia.mid.networkavailability	Network availability. The property return value can contain "available" or "unavailable".
com.nokia.mid.batterylevel	The battery charge level of device. The property value is the actual battery level in percentage.
com.nokia.mid.countrycode	Current network country code.
com.nokia.mid.dateformat	The preferred time format of device user as a string pattern, which is formatted according to the rules defined in Java 2, Standard Edition, version 1.4.1, java.text.SimpleDateFormat class documentation for patterns.
com.nokia.mid.timeformat	The preferred time format of device user as a string pattern, which is formatted according to the rules defined in Java 2, Standard Edition, version 1.4.1, java.text.SimpleDateFormat class documentation for patterns.

## Other

String	Description
com.nokia.mid.dateformat	The preferred date format indicated as a string pattern, which is formatted according to the rules defined in Java™ 2 Platform, Standard Edition, version 1.4.1, java.text.SimpleDateFormat class documentation for date patterns. Only a subset of SimpleDateFormat-defined pattern letters are used in this system property.
com.nokia.mid.timeformat (in Series 40 devices only)	The preferred time format indicated as a string pattern, which is formatted according to the rules defined in Java™ 2 Platform, Standard Edition, version 1.4.1, java.text.SimpleDateFormat class documentation for time patterns. Only a subset of SimpleDateFormat-defined pattern letters are used in this system property.
com.nokia.network.access (in Series 40 devices only)	Returns the networking access point type. the possible values for this property are: <ul style="list-style-type: none"> <li>▪ pd — Packet data, for example GPRS</li> <li>▪ csd — Circuit-switched data, for example GSM CSD/HSCSD data call</li> <li>▪ bt_pan — Bluetooth PAN network</li> <li>▪ na — Not applicable, the implementation cannot determine the type</li> </ul>
com.nokia.mid.imei (in Series 40 3rd Edition devices only, not in Nokia 6270 and Nokia 6111)	Returns the device's IMEI code. Note: The MIDlet must be signed to manufacturer or operator domain; otherwise this value is "null".

## See also

- [How to read system properties in Java ME](#)
- [JDL: System properties](#)

