

# Barcode Contents

[Jump to bottom](#)

Sean Owen edited this page 10 days ago · 14 revisions

Much has been said about 2D barcodes, and the discussion has focused on the format of the 2D barcode itself -- QR Code, Data Matrix, and so on. But equally important is the format of what the barcode itself encodes.

2D barcodes encode text, generally, but that text can represent many things. Commonly, 2D barcodes encode text that represents a URL, like `https://google.com/m`. This is a special string of text since it is recognizable as a URL by readers, and therefore can be acted upon: the reader can open the URL in a browser.

2D barcodes can encode many types of actionable text. Text representing contact information, when recognized, could trigger a prompt to add the contact to an address book. But this only works when readers understand that text encodes contact information. For this, we need standards too.

There are some standards -- de facto and otherwise -- already in use. This wiki attempts to catalog some possible standards for encoding various types of information, and suggest a standard action associated to them. It is not necessarily complete and contributions are welcome.

The ZXing reader library supports all of the formats mentioned in this wiki and a bit more.

## URL

The most common application of barcodes is to encode the text of URL such as `https://google.com/m`. To do so, simply encode exactly the text of the URL in the barcode: `https://google.com/m`. Include the protocol (`https://` here) to ensure it is recognized as a URL.

Readers should open the URL in the device's web browser when decoding a URL. It is probably desirable for a reader to display the URL and ask the user whether to proceed, so that the user may see the URL before accessing it.

URLs prefixed with `URLT0:` have been observed "in the wild" (e.g. `URLT0:google.com/m`).

It's interesting to note that, actually, QR Codes can encode data more efficiently in some cases if only uppercase letters are used. That's because it has a special encoding mode for text consistent of only (uppercase) letters, numbers, and common symbols. It may be advantageous to encode a URL like `HTTP://MYSITE.COM/F00...` rather than `http://mysite.com/foo...` for this reason. However, this depends upon the web server responding to requests correctly when the URI is uppercased. It's not necessarily true that URIs and paths are treated as case-insensitive by a web server, since URIs are technically case sensitive. Don't try this unless you test it to know it works.

Note that NTT DoCoMo uses a [MEBKM bookmark format](#) to express not only a URL but title. This could be treated like a URL, or, prompt the user to add a browser bookmark.

## E-mail address

---

To encode an e-mail address like `sean@example.com`, one could simply encode `sean@example.com`. However to ensure it is recognized as an e-mail address, it is advisable to create a proper `mailto:` URL from the address: `mailto:sean@example.com`.

All of the standard `mailto:` tricks/links should work here as well. Readers should open a blank e-mail message to the given address.

# Address

`mailto:someone@yoursite.com`

# Address, subject

`mailto:someone@yoursite.com?subject=Mail%20from%20our%20Site`

# Address, CC, BCC, subject

`mailto:someone@yoursite.com?`

`cc=someoneelse@theirsite.com,another@thatsite.com,me@mysite.com&bcc=lastperson@the`

# Address, CC, BCC, subject, body

`mailto:someone@yoursite.com?`

`cc=someoneelse@theirsite.com,another@thatsite.com,me@mysite.com&bcc=lastperson@the`

See also...

- [CSS-Tricks: Mailto Links](#)
- [Apple URL Scheme Reference: Mail Links](#)
- [Android: Common Intents](#)
- [RFC 6068](#)

Note that NTT DoCoMo has standardized a more expressive [MATMSG format](#) for encoding an e-mail address, subject, and message.

## Telephone numbers

---

A `tel` URI should be used to encode a telephone number, to ensure that the digits are understood as a telephone number. Further, you should generally use the most complete version of a telephone number possible (i.e., country code + area code + number). For example, to encode the US phone number 212-555-1212, one should encode `tel:+1-212-555-1212`. This tel URI includes a "+1" prefix that will make it usable outside the United States.

Readers should invoke the device's dialer, if applicable, and pre-fill it with the given number, but not automatically initiate a call.

```
# NYC Directory assistance  
tel:+12125551212
```

See Also...

- [CSS-Tricks: The Current State of Telephone Links](#)
- [Apple URL Scheme Reference: Phone Links](#)
- [Android: Common Intents](#)

## Contact information

---

We have, for example, the [vCard](#) format for encoding contact information as text. This format proves a bit verbose for use in 2D barcodes, whose information capacity is limited. It is not clear whether vCard is or should be used to encode contact information.

NTT DoCoMo has popularized a compact [MECARD format](#) for encoding contact information. For example, to encode the name Sean Owen, address "76 9th Avenue, 4th Floor, New York, NY 10011", phone number "212 555 1212", e-mail `srowen@example.com`, one would encode this in a barcode:

```
MECARD:N:Owen,Sean;ADR:76 9th Avenue, 4th Floor, New York, NY  
10011;TEL:12125551212;EMAIL:srowen@example.com;;
```

See the link above for complete information. Note that this format has generally been used with QR Codes only in the past, but, may well be parsed and understood when encoded in a Data Matrix symbol too.

Readers should open a new address book entry, populated with the given data, and prompt the user to add a new contact.

## AU format

Note that KDDI AU also proposes a [slightly different format](#) for this information.

## BIZCARD

We have also observed a BIZCARD format in use but have yet to locate a definitive reference on the format. One can guess its structure from a few examples; it resembles the MECARD format:

```
BIZCARD:N:Sean;X:Owen;T:Software Engineer;C:Google;A:76 9th Avenue, New York, NY
10011;B:+12125551212;E:srowen@google.com;;
```

## vCard

Many apps understand the [vCard specification](#). The wiki for [mangstadt/ez-vcard](#) provides documentation for [Version differences](#) and [Property lists](#) between the various versions of the vCard specification.

```
BEGIN:VCARD
VERSION:3.0
N:Owen;Sean;;;
FN:Sean Owen
TITLE:Software Engineer
EMAIL;TYPE=INTERNET;TYPE=WORK;TYPE=PREF:srowen@google.com
URL;TYPE=Homepage:https://example.com
END:VCARD
```

## SMS/MMS/FaceTime

Much like an e-mail address, one can encode an SMS shortcode or number by creating an `sms` URI. For example to create a link to the number 12345 one would encode `sms:12345`. See [RFC 5724](#) for details.

Likewise `SMSTO:` URLs have been observed in the wild, though I am still looking for an official specification for it. Follow this with the number to SMS. We have heard of URIs of the form `sms:[number]:[subject]`, and likewise for other prefixes like `smsto:`.

There appear to be "mms:" and "MMSTO:" URIs used like "sms:" URIs in practice. We assume the format is the same, and that the reader should react similarly to such a URI.

Readers should open a new SMS message, ready for the user to compose and send it.

```
# Send an SMS/MMS to a number
sms:+18005551212
```

```
# Send an SMS/MMS to a number with pre-filled message.
```

```
sms:+18005551212:This%20is%20my%20text%20message.
```

```
# FaceTime Video
facetime:+18005551212
facetime:me@icloud.com
```

```
# FaceTime Audio
facetime-audio:+18005551212
facetime-audio:me@icloud.com
```

See Also...

- [CSS-Tricks: iPhone Calling and Texting Links](#)
- [Apple URL Scheme Reference: SMS Links](#)
- [Apple URL Scheme Reference: FaceTime Links](#)
- [Android: Common Intents](#)

## Maps, Geographic information

---

A `geo` URI may be used to encode a point on the earth, including altitude. For example, to encode the Google's New York office, which is at 40.71872 deg N latitude, 73.98905 deg W longitude, at a point 100 meters above the office, one would encode `"geo:40.71872,-73.98905,100"`.

A reader might open a local mapping application like Google Maps to this location and zoom accordingly, or could open a link to this location on a mapping web site like Google Maps in the device's web browser.

Services like [Apple Maps](#) and [Google Maps](#) have more thorough implemenations with more options. These are presented as normal HTTPS-based URLs, but are generally handled by the built-in mapping applications.

- [Apple Maps: Space Needle](#)
- [Google Maps: Space Needle](#)

## Calendar Events

---

The newer [iCalendar](#) ( `.ics` ) format, as well as the older [vCalendar](#) ( `.vcs` ) format both define a sub-set of their formats for *events*. These are [vEvents](#).

```
BEGIN:VEVENT
SUMMARY:Summer+Vacation!
DTSTART:20180601T070000Z
DTEND:20180831T070000Z
END:VEVENT
```

## Wi-Fi Network config (Android, iOS 11+)

We propose a syntax like "MECARD" for specifying wi-fi configuration. Scanning such a code would, after prompting the user, configure the device's Wi-Fi accordingly. Example:

```
WIFI:T:WPA;S:mynetwork;P:mypass;;
```

Parameter	Example	Description
T	WPA	Authentication type; can be WEP or WPA or WPA2-EAP , or nopass for no password. Or, omit for no password.
S	mynetwork	Network SSID. Required. Enclose in double quotes if it is an ASCII name, but could be interpreted as hex (i.e. "ABCD" )
P	mypass	Password, ignored if T is nopass (in which case it may be omitted). Enclose in double quotes if it is an ASCII name, but could be interpreted as hex (i.e. "ABCD" )
H	true	Optional. True if the network SSID is hidden. Note this was mistakenly also used to specify phase 2 method in releases up to 4.7.8 / Barcode Scanner 3.4.0. If not a boolean, it will be interpreted as phase 2 method (see below) for backwards-compatibility
E	TTLS	(WPA2-EAP only) EAP method, like TTLS or PWD
A	anon	(WPA2-EAP only) Anonymous identity
I	myidentity	(WPA2-EAP only) Identity
PH2	MSCHAPV2	(WPA2-EAP only) Phase 2 method, like MSCHAPV2

Order of fields does not matter. Special characters \ , ; , , , " and : should be escaped with a backslash ( \ ) as in MECARD encoding. For example, if an SSID was literally "foo;bar\baz" (with double quotes part of the SSID name itself) then it would be encoded like: `WIFI:S:\"foo\;bar\\baz\";;`

## Platform-specific

### Google Play Store

You can construct URIs that (on Android devices) link directly into Google Play Store. For example to encode a link to an app whose package is `org.example.foo` , use:

market://details?id=org.example.foo

## iTunes and App Store

```
# Apple Music, iTunes Music Store
https://geo.itunes.apple.com/us/album/concrete-and-gold/1249068417?
mt=1&app=music
https://geo.itunes.apple.com/us/album/concrete-and-gold/1249068417?
mt=1&app=itunes

# iBooks Store
https://geo.itunes.apple.com/us/book/crazy-rich-asians/id588213164?mt=11

# App Store
https://itunes.apple.com/us/app/fortnite/id1261357853?mt=8
```

See also...

- [Apple URL Scheme Reference: iTunes Links](#)
- [iTunes Link Maker](#)

## YouTube

- [Apple URL Scheme Reference: YouTube Links](#)

▼ Pages 8

Find a Page...

Home

Barcode Contents

Chart Server Parameters

Frequently Asked Questions

Getting Started Developing

License Questions

Scanning From Web Pages

Scanning Via Intent

### Clone this wiki locally

https://github.com/zxing/zxing/wiki.git