


What is an SMS Message?

Message Length, Encoding, and Concatenation

SMS, commonly referred to as “text-messaging” in the USA, is an acronym for **Short Message Service**. An SMS message has [a maximum size of 140 bytes](#). The maximum number of characters in a single SMS message depends on the encoding used, and the encoding used depends on the content of the message.

- A message containing **text characters only** will generally be encoded using **GSM-7, using up 7 bits for 1 character** (some, especially international carrier networks, will use other types of encoding such as 8-bit Latin-1, Latin-9, UTF-8, mostly to include intl. language characters).
- A message containing **special characters, non-Latin characters, emojis, html characters, etc.** will generally be encoded using **UCS-2, using up 16 bits (2byte) for 1 character, drastically reducing the available number of characters per message segment**.

This chart shows the maximum number of characters that can be sent in a **single SMS segment** to carriers.

Message	Type	Characters used in message	Encoding	Max characters/ message (without UDH)
Hello World! How are you?	Text	GSM Standard	GSM-7	160
Have a nice day! 	Text and Emoji	Unicode	UCS-2	70
我係挪威人	Unicode	Unicode	UCS-2	70

How are concatenated (long) messages sent?

Depending on the message content (plain text, emojis, special characters, etc.), carriers/intercarriers will generally use either GSM-7 or UCS-2 encoding to send the messages and each encoding has

limitations to the number of characters that can be sent. The limitations for a single segment are:

- 160 characters for GSM-7 (Latin-1 and UTF-8 support only 140)
- 70 characters for UCS-2 or UTF-16

When you send a message with text longer than the maximum number of characters per segment, the intercarrier network will automatically split the message for you, add a special header (User Data Header), and send multiple SMS messages to the carriers.

What is a User Data Header?

The [User Data Header](#) (UDH) takes up 6 bytes and instructs the receiving device how to reassemble the segments so that your whole message will be shown as one SMS on the receiving handset. The maximum number of characters per concatenated (long) message is slightly reduced due to the inclusion of concatenation headers (UDH).

How to calculate characters and segments

Here are some important units to start with:

- SMS messages are maximum of 140 bytes
- 1 byte = 8 bits
- In GSM-7 encoding, 1 character = 7 bits
- In UCS-2, 1 character = 16 bits
- UDH = 6 bytes

Message Type	Calculation	Max Characters per Segment
GSM Single Segment	$(140 \text{ bytes} \times 8 \text{ bits}) / 7 \text{ bits}$	160 characters
GSM Multi Segment	$(140 \text{ bytes} - 6 \text{ bytes}) \times 8 \text{ bits} / 7 \text{ bits}$	152 characters
Unicode Single Segment	$(140 \text{ bytes} \times 8 \text{ bits}) / 16 \text{ bits}$	70 characters
Unicode Multi Segment	$(140 \text{ bytes} - 6 \text{ bytes}) \times 8 \text{ bits} / 16 \text{ bits}$	67 characters

Examples of message segment counts:

Characters used in message	Total number of characters	Encoding	Message Segments	Calculation
Text Only	160	GSM-7	1	No UDH required, all 160 characters are available
Text Only	240	GSM-7	2	152+88=240 characters
Text and emojis	150	UCS-2	3	67+67+16=150 characters

How can I see how many segments my message will be?

This handy tool, [SMS Split](#), is a great way to check how your message will be encoded and split into segments.

Why does my message look like GSM characters but is being split as a UCS message?

Non-GSM characters often go unnoticed as they look like regular characters. Some prime examples are the "smart quote" (U+2019 to U+0027), a result of text editors trying to be helpful, and the "white space" character (U+2002 to U+0020) that typically surfaces when copying and pasting. The best practice is to use a plain text editor when drafting messages and the [SMS Split](#) tool to determine if you have one of these characters hiding in your message.