

# Control characters in ASCII and Unicode

**Tens of odd control characters appear in ASCII charts. The same characters have found their way to Unicode as well. CR, LF, ESC, CAN... what are all these codes for? Should I care about them? This is an in-depth look into control characters in ASCII and its descendants, including Unicode, ANSI and ISO standards.**

When ASCII first appeared in the 1960s, control characters were an essential part of the new character set. Since then, many new character sets and standards have been published. Computing is not the same either. What happened to the control characters? Are they still used and if yes, for what?

This article looks back at the history of character sets while keeping an eye on modern use. The information is based on a number of standards released by ANSI, ISO, ECMA and The Unicode Consortium, as well as industry practice. In many cases, the standards define one use for a character, but common practice is different. Some characters are used contrary to the standards. In addition, certain characters were originally defined in an ambiguous or loose way, which has resulted in confusion in their use.

## Contents

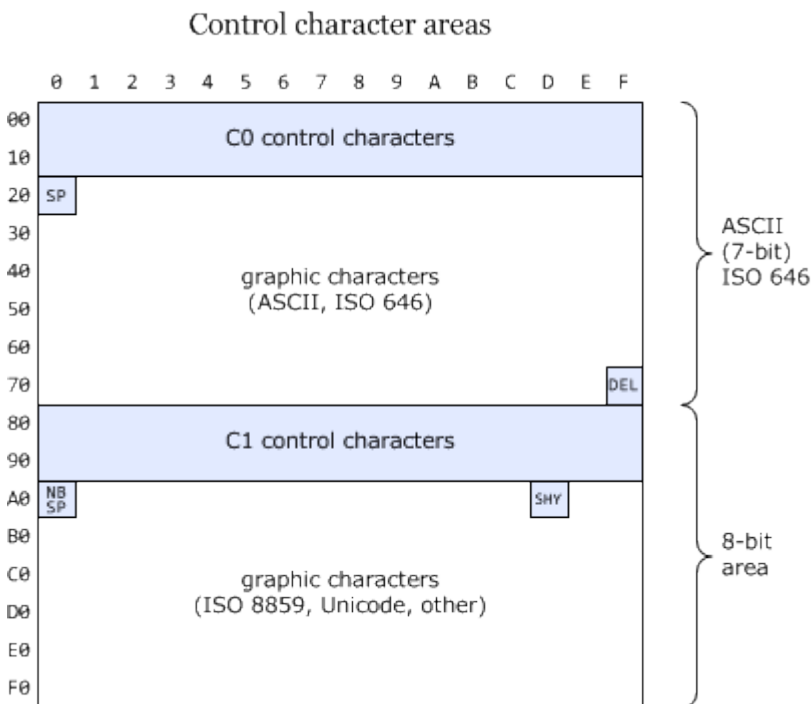
- [Groups of control characters](#)
- [Control characters in standards](#)
  - [ASCII control characters](#)
  - [C1 control characters](#)
  - [ISO 8859 special characters NBSP and SHY](#)
  - [Control characters in Unicode](#)
- [Control characters in modern applications](#)
- [Character list](#)
  - [ASCII](#)
  - [C1](#)
  - [ISO 8859](#)
- [Categories](#)
- [Translations](#)
- [Character index](#)
- [Sources](#)

This article starts by looking at the history of control characters in standards. We then move to modern times. The rest of the article lists all the control characters in detail.

<b>C0</b>	<u>NUL</u> 00	<u>SOH</u> 01	<u>STX</u> 02	<u>ETX</u> 03	<u>EOT</u> 04	<u>ENQ</u> 05	<u>ACK</u> 06	<u>BEL</u> 07	<u>BS</u> 08	<u>HT</u> 09	<u>LF</u> 0A	<u>VT</u> 0B	<u>FF</u> 0C	<u>CR</u> 0D	<u>SO</u> 0E	<u>SI</u> 0F
	<u>DLE</u> 10	<u>DC1</u> 11	<u>DC2</u> 12	<u>DC3</u> 13	<u>DC4</u> 14	<u>NAK</u> 15	<u>SYN</u> 16	<u>ETB</u> 17	<u>CAN</u> 18	<u>EM</u> 19	<u>SUB</u> 1A	<u>ESC</u> 1B	<u>FS</u> 1C	<u>GS</u> 1D	<u>RS</u> 1E	<u>US</u> 1F
	<u>SP</u> 20	.....														<u>DEL</u> 7F
<b>C1</b>	<u>PAD</u> 80	<u>HOP</u> 81	<u>BPH</u> 82	<u>NBH</u> 83	<u>IND</u> 84	<u>NEL</u> 85	<u>SSA</u> 86	<u>ESA</u> 87	<u>HTS</u> 88	<u>HTJ</u> 89	<u>VTS</u> 8A	<u>PLD</u> 8B	<u>PLU</u> 8C	<u>RI</u> 8D	<u>SS2</u> 8E	<u>SS3</u> 8F
	<u>DCS</u> 90	<u>PU1</u> 91	<u>PU2</u> 92	<u>STS</u> 93	<u>CCH</u> 94	<u>MW</u> 95	<u>SPA</u> 96	<u>EPA</u> 97	<u>SOS</u> 98	<u>SGCI</u> 99	<u>SCI</u> 9A	<u>CSI</u> 9B	<u>ST</u> 9C	<u>OSC</u> 9D	<u>PM</u> 9E	<u>APC</u> 9F
<b>8859</b>	<u>NBSP</u> A0														<u>SHY</u> AD	
<b>Delimiter</b>	<b>Introducer</b>	<b>Shift</b>	<b>Format effector</b>	<b>Info separator</b>	<b>Presentation control</b>	<b>Graphic</b>	<b>Area definition</b>	<b>Device control</b>	<b>Transmission control</b>	<b>Misc</b>	<b>Not assigned</b>					

The above clickable table summarizes the control characters. The character codes are given in hexadecimal. Color coding indicates character category. Click a character to jump to more information on it.

## Groups of control characters



For the purposes of this document, control characters are divided into three groups.

**1. ASCII control characters.** The ASCII control character area covers code positions 0–31 (hex 00–1F). This area is also called the C0 set. Two additional controls appear at 32 and 127 (hex 20 and 7F). The ASCII control characters cover a wide range of uses, such as text layout, transmission and device control, and more. [More](#)

**2. C1 control characters.** C1 covers positions 128-159 (hex 80-9F). C1 is primarily for displays and printers. This set is related to ANSI escape sequences and VT100. [More](#)

**3. ISO 8859 special characters.** Two special characters, NBSP and SHY, are from ISO 8859. They are also used in Windows and Unicode. They appear at 160 and 173 (hex A0 and AD). [More](#)

*Note: These control character sets are not the only control characters ever used. Other C0 and C1 sets do exist. Alternative sets were defined for special uses. In them, a part of the standard C0/C1 controls have been deleted or replaced by new controls. Even totally different alternative sets exist. Alternative control characters are not discussed in this article. One can find them in the [International Register of Coded Character Sets](#).*

---

## Control characters in standards

### ASCII control characters

*C0 = positions 0–31. Origin with ASCII and ISO 646 character sets. Characters SP and DEL appear together with C0.*

The first group of control characters originates from ASCII. These characters consist of a set called C0 and two additional characters. The C0 set is in locations 0 to 31. Two additional ASCII characters, SP and DEL, fall outside the C0 area, but they are closely related to the C0 set. All of these characters are defined by the same standards.

This set of control characters covers many uses. There are "Format Effectors" that control the appearance of plain text. There are "Transmission Controls" for use with transmission protocols and "Device Controls" to start, operate and stop auxiliary devices. There are "Information Separators" that delimit various pieces of data. Other controls exist for producing alerts, filling a media, indicating end of media, and for dealing with errors. There are even controls to create new characters and controls. The C0 set was defined with perforated tape, punched cards and typewriter-like devices in mind. Devices have changed since then, but the C0 controls have survived.

### *History of ASCII control characters*

The first version of ASCII was released in 1963. Like the ASCII of today, the 1963 version covered some letters and symbols, as well as control characters. While many of those 35 control characters were similar to those of modern ASCII, some were different. ASCII-1963

had some serious shortcomings, such as no support for lower case letters. It quickly turned out that the standard must be revised. Today, ASCII-1963 is practically forgotten. Since ASCII-1963 deviates a lot from later ASCII versions in the control character area too, we will not go any deeper into it.

The next revision was ASCII-1965. This version, although formally accepted, was not published. Another revision was going to take place. ASCII as we know it is based on the ASCII-1967 standard (USAS X3.4-1967). This version was an important milestone. It was already very close to the version that then became widely used.

In 1968 ASCII was slightly updated and released as USAS X3.4-1968 (later retronamed as ANSI X3.4-1968). The actual updates were very small, only adding an option to use the character LF as a "newline", and designating ASCII and USASCII as the names of the standard. (Later on, the name USASCII was dropped, leaving ASCII as the official name.)

ASCII-1968 became immensely popular. Almost all of today's computer systems use ASCII or one of its descendants. (A notable exception is EBCDIC used on IBM mainframes, very different from ASCII.) The Internet is based on ASCII-1968 as well.

ASCII-1968 defined the 34 control characters that remained: the C0 set, SP and DEL. Included was a short description of the intended functionality of each control character. These definitions also made themselves to [RFC 20](#) word for word. Most of these definitions have remained materially unchanged for decades. Later standards have updated the text, but the basic functionality is still the same. This is what comes to standards. Non-standard use is common and often contrary to the standards.

When ASCII emerged, computing equipment was quite different from the equipment that ASCII was going to be popularized with. Computers were regularly operated through punched cards, perforated tape and teletypewriters (TTYs). TTYs were typewriter-like devices, which were used as interactive computer terminals. Instead of a monitor they produced output on paper. The ASCII control characters were naturally designed considering the devices of those days. Since then, new devices such as monitors have emerged. It hasn't always been that simple to accommodate the control characters to the newer devices. Despite the challenges, the control characters of the 1960s are still with us.

**ISO 646.** ASCII evolved to an ISO standard, which is known as ISO 646. The first version came out in 1967. ISO 646 is the "international edition" of ASCII, with a few differences. Despite the differences, these standards were closely related. ISO 646 allowed national variants to support the national characters required for each country. The US national variant was ASCII. Several other national variants were released to support accented letters (à, ü and the like) and other symbols. The ISO variants including ASCII were a common way to express text in the 1970s and 1980s.

As to the control characters, the ASCII control characters set also appeared in ISO 646. The functionality of the control characters remained quite intact, even though the definitions were updated.

**More standards.** ISO 646 was also released as [ECMA-6](#). The control characters appear in ECMA-6 very similar to those of ISO 646.

A part of the C0 codes were further refined in other standards. SI, SO and ESC appeared as character set extension controls in ANSI X3.41, ISO 2022 and ECMA-35. These characters became widely used to invoke additional character sets. The Transmission Control characters (T<sub>1</sub> to T<sub>10</sub>) appeared as ISO 1745 in 1975, which gave detailed description of where and how they should be used. How widely ISO 1745 was actually used in transmission is another question.

### ***Current status of ASCII control characters***

ASCII was later updated in 1977 and again in 1986 to be in conformance with ISO 646. The control characters in ASCII-1986 and ISO 646/ECMA-6 are very similar, even though minor differences do exist.

The current ISO and ECMA versions, namely ISO 646:1991 and ECMA-6:1991, no longer define the C0 control characters. The control characters didn't go away, however. They now appear in ISO/IEC 6429:1992 and [ECMA-48:1991](#), respectively. Simply put, the C0 set was lumped together with other control characters, the C1 set, which follows below.

As to some specific control characters, the current detailed definitions of SI, SO and ESC can be found in ANSI X3.41, ISO 2022 and ECMA-35. The current details for the Transmission Control characters (TC<sub>1</sub> to TC<sub>10</sub>) appear in the old ISO 1745 from 1975.

Even though the history of the various standards related to the ASCII control codes may sound unnecessarily complicated, the standard functionality of the characters has not changed dramatically. It's still mostly the same as back in 1967. This is what comes to standards. The practice is totally different. Some control characters are indeed commonly used the standard way. On the other hand, many are used contrary to the standards, or simply ignored. It's not uncommon to find control characters forbidden in data. Control characters can have unwanted or unknown side-effects. The easiest way for programmers to deal with them is to shut their eyes or deny such characters altogether.

---

### **C1 control characters**

*C1 = positions 128–159. Primarily for displays and printers.*

The C1 set appeared in the late 1970s. It is primarily designed for controlling display and printer devices, even though some of the controls warrant other uses as well. The C1 set is intended for use with the C0 set.

The C1 set includes "Format effectors" that control horizontal and vertical movement when displaying or printing. There are "Presentation controls" for defining line-break behavior. There are "Area definition" controls for form filling. There are "Introducers" and "Shift Functions" to support extra controls and characters. Additional controls exist for sending command strings and setting an indicator. Some of the controls were intended to cover for shortcomings in the C0 set. Some controls were reserved: 2 controls are for private use, while 4 controls were (and still are) reserved for future standardization.

The C1 set occupies positions 128–159 in 8-bit environments. There are also escape codes to use the C1 set on 7-bit systems. The respective escape codes (ESC *char*) are given in the [C1 list](#) further below.

## ***History of C1***

In 1979 ANSI released additional controls for use with ASCII (ANSI X3.64). This came to be known as the C1 set. A similar set was also released as [ECMA-48](#). According to ANSI, the C1 controls were intended for input/output control of two-dimensional character-imaging devices, including interactive terminals of both the cathode ray tube and printer types, as well as output to microfilm printers.

A bit later, in 1983, the C1 set was standardized as ISO 6429. Standard-wise, the C1 set has been volatile. Both ISO 6429 and ECMA-48 were updated several times. New control characters were added and definitions updated. One of the C1 characters ([IND](#)) was eventually deprecated and removed.

The standards actually cover more control codes than those that fit in the C1 area. These additional controls are used via control sequences (escape sequences). The sequences are beyond the subject of this article. Let it suffice that the sequences are an important part of the standards that should be used together with the C1 controls. The sequences, together with C1, are also known as VT100 and ANSI escape sequences.

## ***Current status of C1***

The current standards for C1 are ISO/IEC 6429:1992 and [ECMA-48:1991](#). These standards now define both the C0 and C1 control characters.

Unicode allows the use of C1 (and C0 too). In fact, the C1 area has been entirely reserved for control codes in Unicode. On the contrary, the (somewhat outdated) DOS and Windows codepages, i.e. character sets, have not reserved space for C1. Instead, they have included additional graphic characters in the C1 area. This doesn't prevent the use of C1 controls on DOS and Windows, though.

In practice, the C1 control characters are not very common. They are specialized codes for special applications.

---

## **ISO 8859 special characters NBSP and SHY**

*Positions 160 and 173.*

ISO 8859 is a group of 8-bit extended character sets. The sets cover various Latin characters and also Cyrillic, Greek, Arabic, Hebrew and Thai characters. ISO 8859 is related to the Windows character sets ("ANSI codepages"), but these are actually different from each other.

Two characters in ISO 8859 are of interest to us: Non-Breaking Space (NBSP) and Soft Hyphen (SHY). They both have control character like properties, even though they are not actually called control characters in ISO 8859.

NBSP appears in position 160 (hex A0) and SHY is 173 (hex AD). The same positions, and roughly the same meanings too, have been adopted to many of the Windows codepages and Unicode.

*Note: ISO 8859-8 Latin/Hebrew defines two additional special characters, namely LRM (left-to-right mark) and RLM (right-to-left mark). These characters are not universal in ISO 8859, but specific to Hebrew. Since LRM and RLM were not used in any other ISO 8859 character set, and since they do not appear in Unicode at the same positions, they are not further presented in this article.*

### ***Current status of NBSP and SHY***

Several current standards include NBSP and SHY. They appear at the same positions in all of the following:

- ISO 8859-1 to 8859-16.  
Exception: ISO 8859-11 Latin/Thai does not include SHY.
- Windows codepages 1250–1258.
- Unicode, block U+0080 C1 Controls and Latin Supplement.

---

### **Control characters in Unicode**

Control characters have made their way to Unicode as well. Unicode recognizes control characters and explicitly allows their use. While Unicode doesn't obsolete control characters, it defines special rules for just a handful of them. Let the standard speak for itself:

The Unicode Standard provides for the intact interchange of these code points, neither adding to nor subtracting from their semantics. The semantics of the control codes are generally determined by the application with which they are used. However, in the absence of specific application uses, they may be interpreted according to the control function semantics specified in ISO/IEC 6429:1992. (Unicode 9.0 p. 822)

Unicode specifies semantics for the following control characters. The semantics appear to be in line with their original semantics, even though some differences may exist.

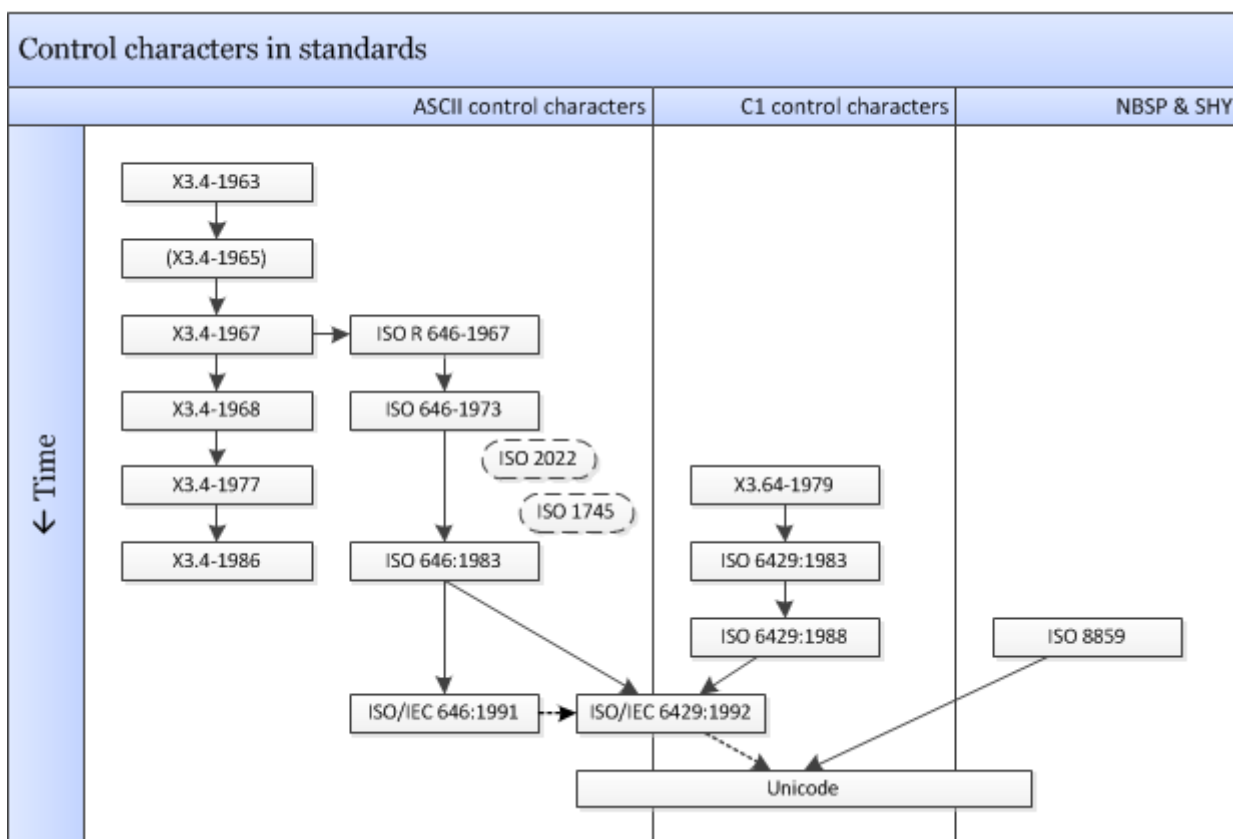
1. ASCII control characters:
  - HT and SP are considered whitespace.
  - LF, VT, FF and CR are considered whitespace, and also mandatory line breaks in the line breaking algorithm.
  - FS, GS, RS and US are considered separators in the bi-directional algorithm.
2. C1 control characters:

- NEL is considered a mandatory line break in the line breaking algorithm, even though supporting it is optional.
3. ISO 8859:
- NBSP and SHY. These characters are not actually control characters in Unicode. Instead, NBSP is "Separator, space" and SHY is "Other, format". Both characters have features in the line-breaking algorithm. With SHY, Unicode is significantly more elaborate than ISO 8859 in that Unicode suggests more hyphenation features than just displaying a hyphen.

*Note: While no new **control** characters appear in Unicode, it does define some of its own special characters, such as formatting characters. These characters are beyond the scope of this article.*

## From ASCII via ISO to Unicode

The following diagram summarizes the development of character standards. You can see how the control characters were propagated from ASCII (X3.4) and other standards to Unicode.



## Control characters in modern applications

With so many control characters coming from the 1960s and 1970s, are they still useful for application programmers?



It depends on the application. Generally speaking, one needs control characters to work with old interfaces or devices. New protocols and file formats tend to use some other mechanism than control characters. Current formats typically use textual markup such as XML, which has little use for control characters beyond whitespace. On the device control side, unless you are writing device drivers, you control devices through operating system calls or library routines rather than sending them control strings to do tricks.

The following is a subjective list of which characters are still in common use and which ones are used less. The list is based on experience writing application software for Windows and DOS.

- ASCII control characters: some used, some not
  - [NUL](#) is still common in everyday use. NUL terminates a string in many programming languages and interfaces.
  - [Transmission control](#) characters (T<sub>1</sub> to T<sub>10</sub>) are generally of little use. Data transfer is done through TCP/IP sockets, HTTP, FTP or some other protocol. Individual transmission control characters appear for special uses.
  - [BEL](#) probably no longer appears in its original use. Rather than sending BEL to produce beeps, applications will rather play a tune via other means.
  - [Format effectors](#) (F<sub>0</sub> to F<sub>5</sub>) are possibly the most important control characters these days. Some of them, such as [CR](#) and [LF](#), are essential for a system to work at all. [HT](#) is also very common, especially in plain text files. [BS](#) and [FF](#) are less common. [VT](#) appears only rarely if ever.
  - [Device control](#) characters (DC<sub>1</sub> to DC<sub>4</sub>) are not required to control devices, really. To control a device from an application you rather make a system call. On the other hand, you might still need XOFF (^S) or XON (^Q) in a command line session from time to time.
  - [SO](#), [SI](#) and [ESC](#) used to be common, but this has changed. One may find them from time to time, but supposedly it's about older systems then.
  - [CAN](#) and [EM](#) are not in common use.
  - [SUB](#) might no longer appear as a substitute. You will more likely see something like "?" or the Unicode REPLACEMENT CHARACTER (U+FFFD) as a substitute for a bad character. Another use for SUB still exists, though. You could find it at the end of a text file.
  - [Information separators](#) (IS<sub>4</sub> to IS<sub>1</sub>) are technically still valid. If anyone uses them to separate information is another question. Other techniques are used instead, such as XML or database systems. As a simple delimiter character a NUL, HT, CR/LF, comma or semicolon is more common than any of the information separators originally designed for the purpose.
  - [SP](#) must be the heaviest used control character of them all.
  - [DEL](#) – well, did you ever see one?
  - Characters ^A to ^Z (1 to 26) frequently appear as keyboard shortcuts in various applications and operating systems. The actual feature triggered by a keyboard shortcut is often unrelated to the respective control character. More of that follows below.
- C1 control characters: little use
  - [NEL](#) is the only C1 character recognized by Unicode. The most probable case to run into NEL is when EBCDIC compatibility is required.
  - The other C1 characters appear outdated now. Since VT100 (that uses C1 extensively) is still a current method with Unix shell sessions, C1 is alive, maybe even everyday business for you. From a programmer's point of view the entire C1 set is rarely used.
- ISO 8859 special characters: in use

- [NBSP](#) is an everyday character to suppress a line break. It is supported by several current standards, including HTML and Unicode.
- [SHY](#) seems to be less frequent.

Some frequently used characters, especially in a special field, may not have been mentioned. If you know frequent current uses for any of the characters, [let us know](#).

Many of the control characters only appear rarely. How did this affect the space efficiency of 7-bit and 8-bit character sets? Instead of reserving space for control characters, it was possible to reuse these areas for additional graphics. This was actually done by DOS, Windows and Mac, all of which assigned graphic characters to the control character areas. Unicode chose to be different in this respect. Since its code space is much larger than 128 or 256, it was possible to reserve the C0/C1 areas entirely for control characters. This has helped the control characters to survive, if not in practical use, then at least in various code charts and lists.

## Keyboards and control characters

Users can create many of the control characters from their keyboards. This usually happens in combination with the `Ctrl` key, and, more rarely, with the `Esc` key. There are also some special keys that produce control characters on their own. `←Backspace`, `Enter`, `Esc`, `Space` and `↵Tab` are the usual ones.

Key presses and control characters, while having some things in common, are usually unrelated. Pressing a key combination doesn't generally trigger the functionality of the respective control character. As an example, while it's possible to press `Ctrl+O` to create an SO (Shift Out), pressing the keys seldom runs the operation associated with SO (pick an alternate character set). Instead, `Ctrl+O` might start an operation beginning with an O, such as "Open".

In some cases a key press does trigger the respective control character feature. Pressing the `↵Tab` key, or `Ctrl+I`, can indeed produce an HT (Horizontal Tabulation) and move the cursor forward on the line. This is an exception rather than the norm, though.

Some key combinations are more likely than others. `Ctrl+A` through `Ctrl+Z` (in other words, `^A` to `^Z`) are common keyboard shortcuts. Control key combinations with a symbol (`^@`, `^[`, `^\`, `^]`, `^^`, `^_`, `^?`) are less common. There is a reason why such combinations should be avoided. Considerable variation exists with symbol keys in different keyboard layouts. A `Ctrl` and symbol key combination doesn't always produce the same control character, or any character at all, which makes it less useful as a keyboard shortcut.

In this article the focus is on the programmatic features of control characters. Less focus is put on the use of keyboard shortcuts.

---

## About the character list

Next we are going to list every control character in detail. The column **Dec** refers to the decimal value of the control code ("ASCII value"). **Hex** is the same in hexadecimal, preceded by a dollar sign for clarity. An octal value is also given. The column **Pos** shows the row/column of the character in code charts.

The list shows key presses that (often) produce the control character on the keyboard. In addition, C-style escape sequences (\c) are provided where available, as are special constants supported by Visual Basic: classic version and Visual Basic .NET.

The last column lists mnemonics and graphic symbols. The symbols (in black) have been standardized, but they have fallen into disuse. The 2-letter mnemonics are standardized for the ASCII section. Additional 2-letter mnemonics for the C1 and ISO 8859 sections are taken from RFC 1345, which is not a standard, but is frequently referred to in this context.

<b>C0</b>	<u>NUL</u> 00	<u>SOH</u> 01	<u>STX</u> 02	<u>ETX</u> 03	<u>EOT</u> 04	<u>ENQ</u> 05	<u>ACK</u> 06	<u>BEL</u> 07	<u>BS</u> 08	<u>HT</u> 09	<u>LF</u> 0A	<u>VT</u> 0B	<u>FF</u> 0C	<u>CR</u> 0D	<u>SO</u> 0E	<u>SI</u> 0F
	<u>DLE</u> 10	<u>DC1</u> 11	<u>DC2</u> 12	<u>DC3</u> 13	<u>DC4</u> 14	<u>NAK</u> 15	<u>SYN</u> 16	<u>ETB</u> 17	<u>CAN</u> 18	<u>EM</u> 19	<u>SUB</u> 1A	<u>ESC</u> 1B	<u>FS</u> 1C	<u>GS</u> 1D	<u>RS</u> 1E	<u>US</u> 1F
	<u>SP</u> 20															
		.....														<u>DEL</u> 7F
<b>C1</b>	<u>PAD</u> 80	<u>HOP</u> 81	<u>BPH</u> 82	<u>NBH</u> 83	<u>IND</u> 84	<u>NEL</u> 85	<u>SSA</u> 86	<u>ESA</u> 87	<u>HTS</u> 88	<u>HTJ</u> 89	<u>VTS</u> 8A	<u>PLD</u> 8B	<u>PLU</u> 8C	<u>RI</u> 8D	<u>SS2</u> 8E	<u>SS3</u> 8F
	<u>DCS</u> 90	<u>PU1</u> 91	<u>PU2</u> 92	<u>STS</u> 93	<u>CCH</u> 94	<u>MW</u> 95	<u>SPA</u> 96	<u>EPA</u> 97	<u>SOS</u> 98	<u>SGCI</u> 99	<u>SCI</u> 9A	<u>CSI</u> 9B	<u>ST</u> 9C	<u>OSC</u> 9D	<u>PM</u> 9E	<u>APC</u> 9F
<b>8859</b>	<u>NBSP</u> A0														<u>SHY</u> AD	
<b>Delim-iter</b>	<b>Intro-ducer</b>	<b>Shift</b>	<b>Format effector</b>	<b>Info separator</b>	<b>Present-ation control</b>	<b>Graphic</b>	<b>Area defi-nition</b>	<b>Device control</b>	<b>Trans-mission control</b>	<b>Misc</b>	<b>Not as-signed</b>					

## Character list

### ASCII control characters (C0)

The ASCII control characters work in 7-bit and 8-bit environments, as well as in Unicode. These controls originate from a set of related standards: ASCII, ISO 646 and ECMA-6, and also ISO 6429 and ECMA-48. All of these characters are available in Unicode, too. The actual C0 set consists of characters NUL through US (0–31). Two additional characters, SP and DEL, are a part of ASCII and the related standards as well.

*\*) The 2-character mnemonics for the ASCII set are from ANSI X3.32, ISO 2047 and ECMA-17. So are also the graphic symbols. The symbols are outdated and rarely used. A couple of the symbols also have alternative forms.*

Dec	Hex	Char	Description	Octal	Pos	*)
0	\$00	NUL	<b>Null</b>	000	0/0	NU
	\0	^@	<p>NUL is defined in the standards as a filler character. It can be used as media-fill or time-fill. NUL doesn't affect the information content of a data stream. It may affect the information layout and the control of equipment, though.</p> <p><i>Note: NUL was originally intended as an ignorable filler character with no meaning. Especially convenient on paper tape, where a NUL equals no holes punched, it could be used to reserve space for new information or correcting errors. ASCII-1986 even suggests NUL as a "time-waster" character to be added after a newline to accommodate mechanical devices where a carriage return works slowly. Despite this, NUL has been used contrary to the standards in null-terminated strings as an End-Of-String marker. Several programming languages use this convention.</i></p> <p>Constant in Visual Basic and VB.NET: vbNullChar, NullChar</p>			☐
1	\$01	SOH	<b>Start of Heading — TC<sub>1</sub> Transmission control character 1</b>	001	0/1	SH
		^A	<p>Indicates the beginning of a heading in a transmission. The heading can be terminated by STX. As per ASCII-1968, a heading constitutes a machine-sensible address or routing information. Later standards have dropped the explanation.</p> <p><i>Note: SOH, along with STX and ETX, was intended for data transmission. It is not intended for marking a heading in a document.</i></p>			┌
2	\$02	STX	<b>Start of Text — TC<sub>2</sub> Transmission control character 2</b>	002	0/2	SX
		^B	<p>STX has two functions in a transmission: it 1) indicates the beginning of a text and 2) may terminate a heading (see SOH). As per ASCII-1968, text is what should be transmitted to a destination. Later standards have dropped the explanation.</p>			└
3	\$03	ETX	<b>End of Text — TC<sub>3</sub> Transmission control character 3</b>	003	0/3	EX
		^C	<p>Terminates a text in a transmission. As per ASCII-1968, a text starts with STX and ends with ETX. Later standards don't necessarily require the pairing of STX with ETX.</p> <p><i>Note: ETX may be used to call for reply from a slave station after a message has been sent. ETX is also commonly used to terminate an interactive process (keyboard: <code>Ctrl+C</code>).</i></p> <p><code>Ctrl+Break</code> on PC keyboard produces this character code.</p>			└
4	\$04	EOT	<b>End of Transmission — TC<sub>4</sub> Transmission control character 4</b>	004	0/4	ET
		^D	<p>Indicates the conclusion of a transmission. The transmission may have contained one or more texts and associated heading(s).</p> <p><i>Note: EOT can be used to end or abort a transmission. It can also be a reply to indicate inability to receive further messages. EOT (keyboard: <code>Ctrl+D</code>) is even used as an End-Of-File control in a Unix shell session.</i></p>			↘

<b>5</b>	<b>\$05 ENQ Enquiry — TC<sub>5</sub> Transmission control character 5</b>	005	0/5	EQ
^E	Requests a response from a remote station. The response may include station identification or status. ENQ can be used as a "Who Are You" (WRU) to identify a remote station, especially after a new connection has been established.			☒ ✚
<b>6</b>	<b>\$06 ACK Acknowledge — TC<sub>6</sub> Transmission control character 6</b>	006	0/6	AK
^F	An affirmative response. Transmitted from a receiver as a response to the sender.  <i>Note: ACK can indicate that a slave station has received a message correctly and is ready to receive more.</i>			✓
<b>7</b>	<b>\$07 BEL Bell</b>	007	0/7	BL
\a ^G	Calls for human attention. BEL may control alarm or attention devices.  <i>Note: BEL is the only control character with an audible effect. It has been used to ring a bell (indeed) or produce a beep sound. A visual alarm is also possible.</i>  <i>In Unicode, this control character is abbreviated BEL but named ALERT, while the name BELL is confusingly used for a graphic character (🔔).</i>			🔔 🔔
<b>8</b>	<b>\$08 BS Backspace — FE<sub>0</sub> Format effector 0</b>	008	0/8	BS
\b ^H	Moves one character position backwards (keeping the previous character).  <i>Note: Contrary to the standards, BS has been used as a combined "move back and delete" operation to remove the previous character. This is not the standard meaning of BS, however. BS is defined as a non-destructive "move back" or "move left" operation, similar to a backspace in mechanical typewriters. To delete the previous character, BS should be followed by DEL. On paper tape the result would be the previous character being completely punched out (erased). BS followed by another character would strike two characters in the same position. Overstriking was a way to produce combined characters. This option was intended to internationalize ASCII. A letter followed by BS followed by a diacritic symbol would produce an accented letter. As an example, u BS ^ would produce û. Several ASCII characters (" ' ` ^ ~ ,) were indeed defined to be used as diacritic symbols. Overstriking could also be suitable with other characters, such as for underlining with the "_" character or printing a slash "/" over "=" to produce "not equal". It could even be used to achieve a strike-through effect (perhaps with -, / or X) to indicate removed text. A boldface effect could be achieved by striking the same character several times at the same position.</i>  <i>Overstriking was a useful option with printing devices, but displays hardly support it. With the advent of more capable character sets and formatting techniques overstriking can be considered outdated. ASCII-1986 does not require overstriking capabilities and suggests that overstriking may be proscribed in the future. ISO 8859 explicitly forbids overstriking.</i>			↶

← Backspace on PC keyboard produces this character code.

Constant in Visual Basic and VB.NET: vbBack, Back

See also: [CCH](#)



<b>9</b>	<b>\$09 HT</b>	<b>Horizontal Tabulation — FE<sub>1</sub> Format effector 1 (Character Tabulation)</b>	011	0/9	HT
----------	----------------	--	-----	-----	----

\t    ^I

Advances to the next pre-determined character position (horizontal tab stop). HT could also be used as a skip function on punched cards. ➤

*Note: HT is commonly also abbreviated TAB.*

*Even though the standards don't set a universal tab width, a typical fixed tab width is 8 columns. Other tab widths, as well as custom tab positions, are used as well. HT is a simple method of data compression: a single character can represent several spaces in formatted text.*

*The  Tab key on the keyboard is consistent with HT in that it usually produces the code HT. How the HT is treated in each application is another story. In windowing environments, there are three common alternative uses. Pressing  Tab can either add an HT character into text, indent text (possibly by adding an appropriate number of spaces or shifting the marginal), or something completely different: jump to the next field or control in a graphical user interface. This way the key has been extended to cover more uses than what HT was originally intended for.*

*The original name of HT is Horizontal Tabulation. It was later renamed as HT Character Tabulation, first in ECMA-48:1986.*

 Tab on PC keyboard produces this character code.

Constant in Visual Basic and VB.NET: vbTab, Tab

<b>10</b>	<b>\$0A LF</b>	<b>Line Feed — FE<sub>2</sub> Format effector 2</b>	012	0/10	LF
-----------	----------------	---	-----	------	----

^J

LF has two alternative functions. It advances to the same character position on the next line (move down), or optionally to the first position on the next line (move to start of next line, i.e. newline). Originally LF was a move-down. A newline option (NL) was added soon. The option allowed LF to be used as a newline, which works like a combined CR LF. Use of LF as a newline requires agreement between sender and recipient of data. Universal agreement has not been reached. ≡

*Note: LF, having two alternative functions, has been a major source of confusion. While LF was initially defined as a "move down" operator, standards began to allow LF as a newline too. As a result, operating systems differ in their definition of a newline. A newline is LF on Unix. Operating systems using CR LF include CP/M, DOS, OS/2 and Windows. Naturally, this caused an incompatibility. To solve the problem, control characters IND and NEL were added to the C1 area. This did not solve the issue, resulting in IND being removed later. ECMA-6:1985 and ASCII-1986 attempted to clarify the situation by declaring LF deprecated for a newline and recommending CR LF instead. ECMA-48:1991 no longer allows LF to function as a newline.*

*The escape sequence for newline and LF is another source of confusion. \n is the common sequence for a newline, whereas there is no such a sequence for a line feed. The actual control character(s) represented by*

`\n` depend on the system. In some cases, `\n` indeed represents LF, but it can also represent another newline sequence.

`Ctrl+Enter` on PC keyboard produces this character code.

Constant in Visual Basic and VB.NET: `vbLf`, `Lf`

See also: [CR](#) [IND](#) [NEL](#)

<b>11</b>	<b>\$0B VT</b>	<b>Vertical Tabulation — FE<sub>3</sub> Format effector 3 (Line Tabulation)</b>	013	0/11	VT
<code>\v</code>	<code>^K</code>	<p>Advances to the same character position on the next pre-determined line. ASCII-1977 and ASCII-1986 optionally allow VT to advance to the first position on the next pre-determined line, if agreed on.</p> <p><i>Note: The original name of VT is Vertical Tabulation. It was later renamed as VT Line Tabulation, first in ECMA-48:1986. VT has been used to jump down to the next pre-defined line when printing on a paper form. According to some sources, vertical tab stops were typically spaced 6 lines apart. VT is a simple data compression method where a single VT represents several LF characters (and optionally a CR too).</i></p> <p><i>In modern use VT must be quite a rare character. As Bob Bemer, one of the original designers of ASCII, put it: "This is a very dangerous character to use. It cannot be used directly on any terminal that I know of. Even if it could, the implementation rules are not supplied unambiguously in the ASCII standard."</i></p> <p>Constant in Visual Basic and VB.NET: <code>vbVerticalTab</code>, <code>VerticalTab</code></p>			⏚
<b>12</b>	<b>\$0C FF</b>	<b>Form Feed — FE<sub>4</sub> Format effector 4</b>	014	0/12	FF
<code>\f</code>	<code>^L</code>	<p>Advances to the next form or page. Standards differ in what column the subsequent character position will be in. Originally, ASCII-1968 did not define the column at all. ISO and ECMA standards declare that FF does not change the column. ASCII-1977 and ASCII-1986 optionally allow, by agreement, moving to the first column, as if FF was actually <code>CF FF</code>.</p> <p><i>Note: FF has been used as "page break" in text files, "new page" on printers and "clear the screen" on displays. The situation was originally unclear whether FF was just a "new page" operator or "new page, move to column 1". ASCII-1977 and ECMA-6:1985 attempted to clarify the situation by recommending the use of <code>CR FF</code>. ASCII-1986 even implied that the "new page, move to column 1" option might be deleted in a future edition of ASCII.</i></p> <p>Constant in Visual Basic and VB.NET: <code>vbFormFeed</code>, <code>FormFeed</code></p>			⏚
<b>13</b>	<b>\$0D CR</b>	<b>Carriage Return — FE<sub>5</sub> Format effector 5</b>	015	0/13	CR
<code>\r</code>	<code>^M</code>	<p>Traditional definition: Moves to the first position on the same line (ASCII, ISO 646, ECMA-6). Newer definition: Moves to the line home position or line limit position of the same line (ISO 6429, ECMA-48).</p> <p><i>Note: The standard meaning of CR is "move to beginning of current line". This allows overprinting the line with new characters, which could be used to achieve underlining, for example. For advancing to the next line CR would be followed by LF. On CP/M, DOS, OS/2 and Windows the</i></p>			⏪

newline marker is CR LF, which is according to the definition. CR alone has been used as the newline character on some systems, such as Commodore and Apple, which use does not conform to the standards in question. The order CR LF (instead of LF CR) may have been important on mechanical devices where a carriage return took relatively long to execute. A non-printing LF was more suitable output while the printing head was returning, rather than striking a graphic symbol in the middle of the line.

Enter on PC keyboard produces this character code.







Constant in Visual Basic and VB.NET: vbCr, Cr

See also: [LF](#)

<b>14</b>	<b>\$0E</b>	<b>SO</b>	<b>Shift Out — LS<sub>1</sub> Locking-Shift One</b>	016	0/14	SO	
		^N	Used to extend the character set. SO may alter the meaning of the following bit combinations until an SI is reached. Between SI and SO, character positions 33-126 (decimal) may represent additional characters that would not otherwise fit in the regular character set.  <i>Note: SO (Shift Out) is normal name of this control. LS1 (Locking-Shift One) is used by ECMA-35 and ECMA-48. In those standards, SO is used in 7-bit environments and LS1 in 8-bit environments. The mechanism to select the alternative character set(s) was defined in ANSI X3.41, ISO 2022 and ECMA-35. It includes the use of escape sequences starting with ESC. SO has also been used on printers to select enlarged characters or another color.</i>				⊗
<b>15</b>	<b>\$0F</b>	<b>SI</b>	<b>Shift In — LS<sub>0</sub> Locking-Shift Zero</b>	017	0/15	SI	
		^0	Used in conjunction with SO. It may reinstate the standard meanings of the characters following it.  <i>Note: SI (Shift In) is normal name of this control. LS0 (Locking-Shift Zero) is used by ECMA-35 and ECMA-48. In those standards, SI is used in 7-bit environments and LS0 in 8-bit environments. SI has also been used on printers to select condensed characters or to reset color.</i>				⊙
<b>16</b>	<b>\$10</b>	<b>DLE</b>	<b>Data Link Escape — TC<sub>7</sub> Transmission control character 7</b>	020	1/0	DL	
		^P	Used to provide supplementary data transmission control functions. DLE changes the meaning of a limited number of following characters.  <i>Note: DLE is the "escape" character for transmission control. DLE can potentially be put in the front of a transmission control character (TC1-TC10) to pass it through "as is" instead of controlling the current transmission. This is not always the case, though. It is possible to create new transmission control sequences with DLE in a similar way ESC is used to create escape sequences for other purposes. Contrary to the standards, ^P has been used as a keyboard shortcut to echo console activity at the printer.</i>				☐
<b>17</b>	<b>\$11</b>	<b>DC<sub>1</sub></b>	<b>Device Control 1 — XON</b>	021	1/1	D1	
		^Q	Intended to turn on or start an ancillary device, to restore it to the basic operation mode (see DC2 and DC3), or for any other device control function.				⌚









*Note: DC1 is conventionally called XON when used in communication for software flow control. The meaning of XON is to continue data transmission after an XOFF (DC3) has been received. The name XON ("transmit on") does not come from a standard, but it is commonly used.*

<b>18</b>	<b>\$12</b>	<b>DC<sub>2</sub></b>	<b>Device Control 2</b>	022	1/2	D2	
		^R	Intended for turning on or starting an ancillary device, set it to a special mode (restored via DC1), or for any other device control function.				
<b>19</b>	<b>\$13</b>	<b>DC<sub>3</sub></b>	<b>Device Control 3 – XOFF</b>	023	1/3	D3	
		^S	Intended for turning off or stopping an ancillary device. It may be a secondary level stop such as wait, pause, stand-by or halt (restored via DC1). It can also perform any other device control function.				
			<i>Note: DC3 is conventionally called XOFF when used in communication for software flow control. An XOFF is issued to stop transmission when a device cannot accept more data. Transmission can be continued via XON (DC1). The name XOFF ("transmit off") does not come from a standard, but it is commonly used. The use of XOFF and XON is in line with the standards, even though not directly defined in them.</i>				
			<i>XOFF (^S) is sometimes used as a pause command. Continuing requires pressing XON (^Q). ^S also works as a pause on MS-DOS and in Windows command prompt. Pressing any key continues.</i>				
<b>20</b>	<b>\$14</b>	<b>DC<sub>4</sub></b>	<b>Device Control 4 (Stop)</b>	024	1/4	D4	
		^T	Intended to turn off, stop or interrupt an ancillary device, or for any other device control function.				
<b>21</b>	<b>\$15</b>	<b>NAK</b>	<b>Negative Acknowledge – TC<sub>8</sub> Transmission control character 8</b>	025	1/5	NK	
		^U	Negative response. Transmitted from a receiver as a response to the sender.				
			<i>Note: NAK can be sent as a response to indicate inability to receive a message, or to request resending.</i>				
<b>22</b>	<b>\$16</b>	<b>SYN</b>	<b>Synchronous Idle – TC<sub>9</sub> Transmission control character 9</b>	026	1/6	SY	
		^V	Used as "time-fill" in synchronous transmission. Sent during an idle condition to retain a signal when there are no other characters to send.				
			<i>Note: SYN has been used by synchronous modems, which have to send data constantly. – Beginning each transmission with at least two SYN characters is a way to achieve synchronization. The receiving station will possibly ignore SYN, since it doesn't belong to the actual data content.</i>				
<b>23</b>	<b>\$17</b>	<b>ETB</b>	<b>End of Transmission Block – TC<sub>10</sub> Transmission control character 10</b>	027	1/7	EB	
		^W	Indicates the end of a block of data. Used when data is divided into blocks for transmission.				

*Note: ETB, when used to end a block, may call for a reply from a slave station.*

<b>24</b>	<b>\$18</b>	<b>CAN</b>	<b>Cancel</b>	030	1/8	CN	
		^X	Indicates that data is in error or should be disregarded. Affects "the data with which it is sent" (ASCII-1968, ASCII-1977) or "the data preceding it" (ASCII-1986, ISO 646, ECMA-6, ECMA-48).				⊗
			<i>Note: There are 2 alternative definitions for the data to be disregarded. The actual scope of cancellation is undefined by the standards and should be defined case by case. ^X has been used as a keyboard shortcut to cancel (delete) the characters on the current line, which use conforms to the standards.</i>				
<b>25</b>	<b>\$19</b>	<b>EM</b>	<b>End of Medium</b>	031	1/9	EM	
		^Y	Identifies 1) the physical end of a medium, 2) the end of the used portion of a medium, or 3) the end of wanted data on a medium.				↓
			<i>Note: EM may have been suitable for paper tape or magnetic tape to say "no more data". Disk file systems use more sophisticated ways to keep track of the used and unused areas of the medium.</i>				
			<i>This character is commonly abbreviated EM, except for Unicode, which provides it as an alias with abbreviation EOM.</i>				
<b>26</b>	<b>\$1A</b>	<b>SUB</b>	<b>Substitute</b>	032	1/10	SB	
		^Z	Used in place of an invalid or erroneous character. Introduced by automatic means in cases like a transmission error.				?
			<i>Note: When SUB is used as a substitution character, the reverse question mark symbol seems quite good as its visual representation. Compare SUB to Unicode U+FFFD REPLACEMENT CHARACTER.</i>				
			<i>SUB has often been used contrary to the standards. On CP/M and MS-DOS, it appears as an End-Of-File marker for text files (^Z). On Unix, ^Z is a keyboard signal to interrupt a foreground process.</i>				
<b>27</b>	<b>\$1B</b>	<b>ESC</b>	<b>Escape</b>	033	1/11	EC	
		\e ^[	The first character of an escape sequence. Provides either supplementary characters or additional control functions. ESC changes the meaning of a limited number of following characters.				⊖
			<i>Note: ESC is used to form escape sequences, which perform various control functions or apply additional character sets. ESC can also be used to invoke the C1 control characters on a 7-bit system that only support character positions 0-127.</i>				
			<i>On the keyboard, sometimes the <b>Esc</b> key indeed produces the ESC control character. In windowing environments, the key typically cancels a dialog or an operation, rather than producing a control character or starting an escape sequence. This kind of an "escape" is not based on the character standards, however. The closest ASCII equivalent for canceling a dialog would be CAN, but since there is no <b>Can</b> key on the common keyboards, it can't be used.</i>				
			<b>Esc</b> on PC keyboard produces this character code.				

<b>28</b>	<b>\$1C FS</b>	<b>File Separator — IS<sub>4</sub> Information separator 4</b>	034	1/12	FS	
	^\	The four information separators (FS, GS, RS and US) are used to separate and qualify data. Each separator has two alternative names: Information Separator Four equals File Separator, Information Separator Three equals Group Separator, Information Separator Two equals Record Separator and Information Separator One equals Unit Separator. The separators can be used either hierarchically or in a non-hierarchical manner. When used hierarchically, the order is US (least inclusive), RS, GS and FS (most inclusive). The content and length of a file, group, record or unit are not specified by the standards.  FS, when used in a hierarchical order, delimits a data item called a file. It can also delimit anything else.				
<b>29</b>	<b>\$1D GS</b>	<b>Group Separator — IS<sub>3</sub> Information separator 3</b>	035	1/13	GS	
	^]	GS, when used in a hierarchical order, delimits a data item called a group. It can also delimit anything else.				
<b>30</b>	<b>\$1E RS</b>	<b>Record Separator — IS<sub>2</sub> Information separator 2</b>	036	1/14	RS	
	^^	RS, when used in a hierarchical order, delimits a data item called a record. It can also delimit anything else.				
<b>31</b>	<b>\$1F US</b>	<b>Unit Separator — IS<sub>1</sub> Information separator 1</b>	037	1/15	US	
	^_	US, when used in a hierarchical order, delimits a data item called a unit. It can also delimit anything else.  <i>Note: The information separators were deliberately arranged next to SPACE, which can also be used as an information separator (word separator).</i>				
<b>32</b>	<b>\$20 SP</b>	<b>Space</b>	040	2/0	SP	
		Moves one character position forwards. Space may also have a function equivalent to that of an information separator.   <i>Note: Space has a dual nature. It can be classified as both a control character and a (non-printing) graphic character. SP is similar to a Format Effector. It can also be used as a fifth Information Separator. Space is sometimes represented by the symbol ¯ or ¨ (b with a stroke) or ◻ (open box). SP does not belong to the C0 set.</i>  Spacebar on PC keyboard produces this character code.  See also: <a href="#">NBSP</a>				
<b>127</b>	<b>\$7F DEL</b>	<b>Delete</b>	177	7/15	DT	
	^?	Outdated. An ignorable character originally intended for erasing an erroneous or unwanted character in punched tape. In this standard use, DEL wouldn't affect the information content of data, even though it may have affected the information layout and the control of equipment. Standards also allowed DEL to be used as media-fill or time-fill (even though a NUL may be more appropriate).  <i>Note: DEL is now outdated. It was removed from the latest standards (ECMA-48 in 1991 and ISO 6429 in 1992). The origin of DEL is with perforated paper. On that, DEL was equal to "all holes punched", which</i>				

is a way to invalidate an erroneous character (rubout). In a sense, DEL is similar to NUL, since both characters mean "nothing". ASCII-1977 suggests the use of DEL as a "time waster" to accommodate mechanical devices where a carriage return takes time to execute. ASCII-1986 recommends NUL as a time waster instead of DEL. DEL does not belong to the C0 set, but is an individual control code.

Ctrl+ ←Backspace on PC keyboard produces this character code.

See also: [NUL](#)

\x is what you write in a C program to produce the given control character. ^x means you press Ctrl+X to produce the given control character.

## C1 control characters

The C1 control characters work in 8-bit environments. These controls come from 3 related standards: ANSI X3.64, ISO 6429 and ECMA-48. All of these characters are also available in Unicode, too. There are three unassigned control characters: PAD, HOP and SGCI. Use was planned for them in a failed draft DIS 10646, but they were not actually standardized or put to use. Despite this, one can find these control characters in various C1 lists online, and also as aliases in later Unicode standards.

†) The 2-character mnemonics for C1 are from RFC 1345. They are not standardized.

Dec	Hex	Char	Description	Octal	Pos	†)
128	\$80	PAD	unassigned, "Padding Character"	200	8/0	PA
ESC @ A reserved control code. Intended for use as PAD Padding Character in draft DIS 10646, rejected, never standardized (not accepted to ISO 10646). <i>Note: Not part of ISO/IEC 6429 or ECMA-48.</i> <i>Unicode lists this character as XXX and provides PAD as an alias.</i>						
129	\$81	HOP	unassigned, "High Octet Preset"	201	8/1	HO
ESC A A reserved control code. Intended for use as HOP High Octet Preset in draft DIS 10646, rejected, never standardized (not accepted to ISO 10646). <i>Note: Not part of ISO/IEC 6429 or ECMA-48. Listed as XXX in Unicode.</i> <i>Unicode lists this character as XXX and provides HOP as an alias.</i>						
130	\$82	BPH	Break Permitted Here	202	8/2	BH
ESC B A point where a line break may occur. <i>Note: Roughly equivalent to a soft hyphen except that the means for indicating a line break is not necessarily a hyphen. Compare to Unicode U+200B ZERO WIDTH SPACE.</i>						
131	\$83	NBH	No Break Here	203	8/3	NH
ESC C A point where a line break may not occur. <i>Note: Compare to Unicode U+2060 WORD JOINER.</i>						

<b>132 \$84 IND</b>	<b>Index</b>	204	8/4	IN
ESC D	Moves to the next line keeping the current horizontal position.			
	<i>Note: According to ECMA-48:1986, IND was provided for use in those cases where LF was implemented as New Line. IND was deprecated in 1988 and withdrawn in 1992 from ISO/IEC 6429 (1986 and 1991 respectively for ECMA-48).</i>			
	See also: <a href="#">LF RI</a>			
<b>133 \$85 NEL</b>	<b>Next Line</b>	205	8/5	NL
ESC E	Moves to the first position of the next line. Alternatively, to line home or line limit position.			
	<i>Note: NEL maps to the control character NL (New Line) in the EBCDIC character set used on IBM mainframes.</i>			
	See also: <a href="#">LF</a>			
<b>134 \$86 SSA</b>	<b>Start of Selected Area</b>	206	8/6	SA
ESC F	Starts a string of character positions whose contents can be transmitted. The string ends at EPA (or end of display).			
<b>135 \$87 ESA</b>	<b>End of Selected Area</b>	207	8/7	ES
ESC G	Ends a string of character positions (started by SPA) whose contents can be transmitted.			
<b>136 \$88 HTS</b>	<b>Horizontal Tabulation Set, Character Tabulation Set</b>	210	8/8	HS
ESC H	Sets a tab stop at the active position.			
	<i>Note: ISO 6429:1992, ECMA-48:1986 and ECMA-48:1991 have renamed HTS as Character Tabulation Set.</i>			
<b>137 \$89 HTJ</b>	<b>Horizontal Tabulation with Justification, Character Tabulation with Justification</b>	211	8/9	HJ
ESC I	Moves text to the following tab stop. The text is what comes after the previous tab stop up to the active position.			
	<i>Note: This character has several names. ANSI X3.64 originally called it Horizontal Tabulation with Justify. ISO 6429:1992, ECMA-48:1986 and ECMA-48:1991 have renamed HTJ as Character Tabulation with Justification.</i>			
<b>138 \$8A VTS</b>	<b>Vertical Tabulation Set, Line Tabulation Set</b>	212	8/10	VS
ESC J	Sets a vertical tab stop at the active line.			
	<i>Note: ISO 6429:1992, ECMA-48:1986 and ECMA-48:1991 have renamed VTS as Line Tabulation Set.</i>			
<b>139 \$8B PLD</b>	<b>Partial Line Down, Partial Line Forward</b>	213	8/11	PD
ESC K	Moves down so that following characters will appear as subscripts. Subscripts end at the next PLU.			
	<i>Note: ISO 6429:1992 and ECMA-48:1991 have renamed PLD as Partial Line Forward. Sample: text PLD subscript PLU text.</i>			
<b>140 \$8C PLU</b>	<b>Partial Line Up, Partial Line Backward</b>	214	8/12	PU

ESC L Moves up so that following characters will appear as superscripts. Superscripts end at the next PLD.

*Note: ISO 6429:1992 and ECMA-48:1991 have renamed PLU as Partial Line Backward. Sample: text PLU superscript PLD text.*

---

<b>141 \$8D RI</b>	<b>Reverse Index, Reverse Line Feed</b>	215	8/13	RI
--------------------	---	-----	------	----

ESC M Moves to the previous line keeping the current horizontal position.

*Note: ISO 6429:1992, ECMA-48:1986 and ECMA-48:1991 renamed RI as Reverse Line Feed, apparently related to the removal of IND.*

See also: [IND](#)

---

<b>142 \$8E SS2</b>	<b>Single Shift Two</b>	216	8/14	S2
---------------------	-------------------------	-----	------	----

ESC N Used to extend the character set. The next character will be from the currently chosen G2 set.

*Note: For more information see ISO 2022 or ECMA-35. The next character should be in the decimal range 33-126 or 32-127.*

---

<b>143 \$8F SS3</b>	<b>Single Shift Three</b>	217	8/15	S3
---------------------	---------------------------	-----	------	----

ESC O Used to extend the character set. The next character will be from the currently chosen G3 set.

*Note: For more information see ISO 2022 or ECMA-35. The next character should be in the decimal range 33-126 or 32-127.*

---

<b>144 \$90 DCS</b>	<b>Device Control String</b>	220	9/0	DC
---------------------	------------------------------	-----	-----	----

ESC P Starts a device control string. ST ends the string. The control string may include commands to the receiving device, or a status report from the sending device.

---

<b>145 \$91 PU1</b>	<b>Private Use One</b>	221	9/1	P1
---------------------	------------------------	-----	-----	----

ESC Q Reserved for private use, no standardized meaning.

---

<b>146 \$92 PU2</b>	<b>Private Use Two</b>	222	9/2	P2
---------------------	------------------------	-----	-----	----

ESC R Reserved for private use, no standardized meaning.

---

<b>147 \$93 STS</b>	<b>Set Transmit State</b>	223	9/3	TS
---------------------	---------------------------	-----	-----	----

ESC S Notifies that data is ready for transfer from a device (ANSI X3.64), or establishes the transmit state in the receiving device (ISO 6429, ECMA-48). Doesn't initiate the actual transmission.

---

<b>148 \$94 CCH</b>	<b>Cancel Character</b>	224	9/4	CC
---------------------	-------------------------	-----	-----	----

ESC T Ignore the preceding graphic character (and CCH itself too). If the previous character is a control character or sequence, ANSI X3.64 says it should be ignored, while ISO 6429 and ECMA-48 leave the action undefined.

*Note: Destructive backspace. Intended to eliminate ambiguity about the meaning of BS.*

See also: [BS](#)

---

<b>149 \$95 MW</b>	<b>Message Waiting</b>	225	9/5	MW
--------------------	------------------------	-----	-----	----

ESC U Sets a message waiting indicator in the receiving device.

<b>150 \$96 SPA</b>	<b>Start of Guarded Protected Area, Start of Protected Area, Start of Guarded Area</b>	226	9/6	SG
ESC V	Starts a string of character positions that can't be altered manually or transmitted. Optionally protects against erasure too. EPA will end the string.			
	<i>Note: SPA is known as Start of Protected Area (ANSI X3.64, ECMA-48:1979), Start of Guarded Protected Area (ISO 6429:1983, ECMA-48:1984) and Start of Guarded Area (ISO 6429:1992, ECMA-48:1986 and ECMA-48:1991).</i>			
<b>151 \$97 EPA</b>	<b>End of Guarded Protected Area, End of Protected Area, End of Guarded Area</b>	227	9/7	EG
ESC W	Ends the area started by SPA.			
	<i>Note: EPA is known as End of Protected Area (ANSI X3.64, ECMA-48:1979), End of Guarded Protected Area (ISO 6429:1983, ECMA-48:1984) and End of Guarded Area (ISO 6429:1992, ECMA-48:1986 and ECMA-48:1991).</i>			
<b>152 \$98 SOS</b>	<b>Start of String</b>	230	9/8	SS
ESC X	Starts a control string. The string ends at ST. It cannot contain a SOS. The interpretation of the string depends on the application.			
<b>153 \$99 SGCI unassigned, "Single Graphic Character Introducer"</b>		231	9/9	GC
ESC Y	A reserved control code. Intended for use as SGCI Single Graphic Character Introducer in draft DIS 10646, rejected, never standardized (not accepted to ISO 10646).			
	<i>Note: Not part of ISO/IEC 6429 or ECMA-48. Listed as XXX in Unicode. Unicode lists this character as XXX and provides SGC as an alias.</i>			
<b>154 \$9A SCI</b>	<b>Single Character Introducer</b>	232	9/10	SC
ESC Z	A reserved control code. The name was standardized as SCI Single Character Introducer, but the actual functionality was not implemented in the standards.			
	<i>Note: SCI was to be followed by a single byte, which would represent a control function or a graphic character. The functions or characters were not defined in the standards.</i>			
<b>155 \$9B CSI</b>	<b>Control Sequence Introducer</b>	233	9/11	CI
ESC [	Starts a control sequence.			
<b>156 \$9C ST</b>	<b>String Terminator</b>	234	9/12	ST
ESC \	Closes a string opened by APC, DCS, OSC, PM or SOS.			
<b>157 \$9D OSC</b>	<b>Operating System Command</b>	235	9/13	OC
ESC ]	Starts an operating system control string. The string ends at ST and is interpreted subject to the operating system.			
<b>158 \$9E PM</b>	<b>Privacy Message</b>	236	9/14	PM
ESC ^	Starts a privacy message. ST will end the message.			
<b>159 \$9F APC</b>	<b>Application Program Command</b>	237	9/15	AC

ESC \_ Starts an application program command string. ST will end the command. The interpretation of the command is subject to the program in question.

ESC X means you press Esc followed by X to produce this control character.

## ISO 8859 special characters

The two special characters, NBSP and SHY, are not really control characters. They are graphic characters with a special feature. The characters also appear in Unicode. They are included here for the sake of completeness.

‡) The 2-character mnemonics for NBSP and SHY are from RFC 1345. They are not standardized.

Dec	Hex	Char	Description	Octal	Pos	‡)
160	\$A0	NBSP	<b>No-Break Space</b>	240	10/0	NS
A space for use when a line break is to be prevented.						
<i>Note: NBSP can sometimes be produced by pressing Ctrl+Shift+SPACE. No universally supported key combination exists.</i>						
<i>In HTML you can write &amp;nbsp; or &amp;#160; to add a no-break space to a web page.</i>						
See also: <a href="#">SP</a>						
173	\$AD	SHY	<b>Soft Hyphen</b>	255	10/13	--
Indicates an intraword break point for use when a word must be broken across lines. The visual rendering either is a hyphen (ISO 8859) or varies (Unicode).						
<i>Note: SHY can sometimes be produced by pressing Ctrl+-. No universally supported key combination exists.</i>						
<i>In HTML you can write &amp;shy; or &amp;#173; to add a soft hyphen to a web page.</i>						



C0	<u>NUL</u> 00	<u>SOH</u> 01	<u>STX</u> 02	<u>ETX</u> 03	<u>EOT</u> 04	<u>ENQ</u> 05	<u>ACK</u> 06	<u>BEL</u> 07	<u>BS</u> 08	<u>HT</u> 09	<u>LF</u> 0A	<u>VT</u> 0B	<u>FF</u> 0C	<u>CR</u> 0D	<u>SO</u> 0E	<u>SI</u> 0F
	<u>DLE</u> 10	<u>DC1</u> 11	<u>DC2</u> 12	<u>DC3</u> 13	<u>DC4</u> 14	<u>NAK</u> 15	<u>SYN</u> 16	<u>ETB</u> 17	<u>CAN</u> 18	<u>EM</u> 19	<u>SUB</u> 1A	<u>ESC</u> 1B	<u>FS</u> 1C	<u>GS</u> 1D	<u>RS</u> 1E	<u>US</u> 1F
	<u>SP</u> 20															
		.....														<u>DEL</u> 7F
C1	<u>PAD</u> 80	<u>HOP</u> 81	<u>BPH</u> 82	<u>NBH</u> 83	<u>IND</u> 84	<u>NEL</u> 85	<u>SSA</u> 86	<u>ESA</u> 87	<u>HTS</u> 88	<u>HTJ</u> 89	<u>VTS</u> 8A	<u>PLD</u> 8B	<u>PLU</u> 8C	<u>RI</u> 8D	<u>SS2</u> 8E	<u>SS3</u> 8F
	<u>DCS</u> 90	<u>PU1</u> 91	<u>PU2</u> 92	<u>STS</u> 93	<u>CCH</u> 94	<u>MW</u> 95	<u>SPA</u> 96	<u>EPA</u> 97	<u>SOS</u> 98	<u>SGCI</u> 99	<u>SCI</u> 9A	<u>CSI</u> 9B	<u>ST</u> 9C	<u>OSC</u> 9D	<u>PM</u> 9E	<u>APC</u> 9F
8859	<u>NBSP</u> A0													<u>SHY</u> AD		
Delimiter	Introducer	Shift	Format effector	Info separator	Presentation control	Graphic	Area definition	Device control	Transmission control	Misc	Not assigned					

## Categories

A summary of character categories. Mostly based on ANSI X3.64, ISO 6429, ECMA-6 and ECMA-48.

### Delimiters (Control string delimiters)

Delimiters start and end a control string. A control string consists of an opening delimiter, a command string or a character string, and a terminating delimiter (ST).

[APC](#) [DCS](#) [OSC](#) [PM](#) [SOS](#) [ST](#)

### Introducers

An introducer is a control character or escape sequence that begins a sequence. The sequence is interpreted as a single graphic character or control.

[CSI](#) [ESC](#) [SCI](#)

### Shift function characters (was: Code extension control)

Shift function characters are used to extend the character set of the code. They may alter the meaning of one or more characters that follow them.

[SI](#) [SO](#) [SS2](#) [SS3](#)

### Format effectors (also: Layout characters)

Format effectors are mainly intended for the control of the layout and positioning of information. Format effectors (most of them) are data which happen to have a format representation rather than a graphic representation.

[BS](#) [CR](#) [FF](#) [HT](#) [HTJ](#) [HTS](#) [IND](#) [LF](#) [NEL](#) [PLD](#) [PLU](#) [RI](#) [VT](#) [VTS](#)

## Information separators

Information separators separate and qualify data logically. They may be used either in hierarchical order or non-hierarchically. Their specific meanings depend on the application.

[FS](#) [GS](#) [RS](#) [US](#)

## Presentation control characters

Presentation control characters indicate where a line break may or may not occur.

[BPH](#) [NBH](#)

## Graphic characters

Graphic characters appearing here are those that have control character like properties.

[NBSP](#) [SHY](#) [SP](#)

## Area definition characters (was: Form filling)

Area definition characters are used for entering information into a preformatted visual display.

[EPA](#) [ESA](#) [SPA](#) [SSA](#)

## Device control characters

Device control characters are intended for the control of local or remote or ancillary devices. They are not intended to control data communication systems; this should be done with transmission control characters.

[DC1](#) [DC2](#) [DC3](#) [DC4](#)

## Transmission control characters (was: Communication control)

Transmission control characters are intended to control or facilitate transmission of information over telecommunication networks.

[ACK](#) [DLE](#) [ENQ](#) [EOT](#) [ETB](#) [ETX](#) [NAK](#) [SOH](#) [STX](#) [SYN](#)

## Miscellaneous

Miscellaneous control characters fall outside other categories.

[BEL](#) [CAN](#) [CCH](#) [DEL](#) [EM](#) [MW](#) [NUL](#) [PU1](#) [PU2](#) [STS](#) [SUB](#)

## Not assigned

Unassigned control characters are ones that were not standardized. Their location was reserved for future standardization. These characters are known by names that appeared in a draft (DIS 10646), even though they didn't make it to the final standard.

[PAD](#) [HOP](#) [SGCI](#)

---

## Translations

The translated terms are taken from the given standards. Several alternative translations may exist.

<b>English</b> <i>Standard</i>	<b>French</b> <i>Unicode 5.0</i>	<b>Russian</b> <i>GOST 34.301-91, GOST 34.302.2-91</i>	<b>Spanish</b> <i>T.53, T.51, T.50</i>	<b>German</b> <i>DIN 66003</i>	<b>Finnish</b> <i>SFS 4017 *</i>
<a href="#">NUL</a> Null	<a href="#">NUL</a> nul	<a href="#">ПУС</a> пусто	nulo	Füllzeichen	tyhjämerkki
<a href="#">SOH</a> Start of Heading	<a href="#">DET</a> début d'en-tête	<a href="#">H3</a> начало заголовка	comienzo de encabezamiento	Anfang des Kopfes	otsikon alku
<a href="#">STX</a> Start of Text	<a href="#">DTX</a> début de texte	<a href="#">HT</a> начало текста	comienzo de texto	Anfang des Textes	tekstin alku

<a href="#">ETX</a>	End of Text	<a href="#">FTX</a>	fin de texte	<a href="#">КТ</a>	конец текста	fin de texto	Ende des Textes	tekstin loppu
<a href="#">EOT</a>	End of Transmission	<a href="#">FTR</a>	fin de transmission	<a href="#">КП</a>	конец передачи	fin de transmisión	Ende der Übertragung	tekstin loppu
<a href="#">ENQ</a>	Enquiry	<a href="#">DEM</a>	demande	<a href="#">КТМ</a>	кто там?	pregunta	Stationsauf-forderung	kysely
<a href="#">ACK</a>	Acknowledge	<a href="#">ACC</a>	accusé de réception [positif]	<a href="#">ДА</a>	подтверждение	acuse de recibo	Positive Rück-meldung	kuittaus
<a href="#">BEL</a>	Bell	<a href="#">SON</a>	sonnerie	<a href="#">ЗВ</a>	звонок	timbre	Klingel	äänimerkki
<a href="#">BS</a>	Backspace	<a href="#">EFF</a>	espace arrière	<a href="#">ВШ</a>	возврат на шаг	retroceso	Rückwärts-schritt	peruutus
<a href="#">HT</a>	Horizontal Tabulation	<a href="#">TAB</a>	tabulation horizontale	<a href="#">ГТ</a>	горизонтальная табуляция	tabulación de caracteres	Horizontal-Tabulator	sarake-ohjaus
<a href="#">LF</a>	Line Feed	<a href="#">PAL</a>	changement de ligne	<a href="#">ПС</a>	перевод строки	cambio de renglón	Zeilen-vorschub	riviaskel
<a href="#">VT</a>	Vertical Tabulation	<a href="#">TAV</a>	tabulation verticale	<a href="#">ВТ</a>	вертикальная табуляция	tabulación vertical	Vertikal-Tabulator	rivitys
<a href="#">FF</a>	Form Feed	<a href="#">SDP</a>	saut de page, page suivante	<a href="#">ПФ</a>	перевод формата	página siguiente	Formular-vorschub	sivun vaihto
<a href="#">CR</a>	Carriage Return	<a href="#">RC</a>	retour de chariot	<a href="#">ВК</a>	возврат каретки	retorno del carro	Wagen-rücklauf	vaunun palautus
<a href="#">SO</a>	Shift Out	<a href="#">HC</a>	hors code	<a href="#">ВЫХ</a>	выход	cambio-salida	Dauerum-schaltung	koodin-vaihto
<a href="#">SI</a>	Shift In	<a href="#">EC</a>	en code	<a href="#">ВХ</a>	вход	cambio-entrada	Rückschaltung	koodin-palautus
<a href="#">DLE</a>	Data Link Escape	<a href="#">ÉCT</a>	échappement transmission	<a href="#">AP1</a>	авторегистр один	escape de enlace de datos	Datenüber-tragungs-umschaltung	ohjaus-koodin poikkeus
<a href="#">DC<sub>1</sub></a>	Device Control 1	<a href="#">CD1</a>	commande d'appareil un	<a href="#">СУ1</a>	символ устройства один	control de dispositivo uno	Geräte-steuerung 1	laitteen ohjaus 1
<a href="#">DC<sub>2</sub></a>	Device Control 2	<a href="#">CD2</a>	commande d'appareil deux	<a href="#">СУ2</a>	символ устройства два	control de dispositivo dos	Geräte-steuerung 2	laitteen ohjaus 2
<a href="#">DC<sub>3</sub></a>	Device Control 3	<a href="#">CD3</a>	commande d'appareil trois	<a href="#">СУ3</a>	символ устройства три	control de dispositivo tres	Geräte-steuerung 3	laitteen ohjaus 3
<a href="#">DC<sub>4</sub></a>	Device Control 4 (Stop)	<a href="#">CD4</a>	commande d'appareil quatre	<a href="#">СУ4</a>	символ устройства четыре	control de dispositivo cuatro	Geräte-steuerung 4	laitteen ohjaus 4
<a href="#">NAK</a>	Negative Acknowledge	<a href="#">ACN</a>	accusé de réception négatif	<a href="#">НЕТ</a>	отрицание	acuse de recibo negativo	Negative Rückmeldung	kielteinen kuittaus
<a href="#">SYN</a>	Synchronous Idle	<a href="#">SYN</a>	synchronisation	<a href="#">СИН</a>	синхронизация	reposo síncrono	Synchroni-sierung	tahditus
<a href="#">ETB</a>	End of Transmission Block	<a href="#">FBT</a>	fin de bloc de transmission	<a href="#">КБ</a>	конец блока	fin de bloque de transmisión	Ende des Übertragungs-blocks	jakson-siirron loppu
<a href="#">CAN</a>	Cancel	<a href="#">ANN</a>	annulation	<a href="#">ОТМ</a>	отмена	cancelar	Ungültig	sanoman peruutus
<a href="#">EM</a>	End of Medium	<a href="#">FS</a>	fin de support	<a href="#">КН</a>	конец носителя	fin del medio físico	Ende der Auf-zeichnung	tieto-välineen loppu
<a href="#">SUB</a>	Substitute	<a href="#">SUB</a>	substitution	<a href="#">ЗС</a>	замена символа	substituto	Substitution	korvike
<a href="#">ESC</a>	Escape	<a href="#">ÉCH</a>	échappement	<a href="#">AP2</a>	авторегистр два	escape	Umschaltung	koodin poikkeus
<a href="#">FS</a>	File Separator	<a href="#">SF</a>	séparateur de fichiers	<a href="#">РФ</a>	разделитель файлов	separador de fichero	Haupt-gruppen-Trennung	tiedoston erotus-merkki

<a href="#">GS</a>	Group Separator	<a href="#">SG</a>	séparateur de groupes	<a href="#">PГ</a>	разделитель групп	separador de grupo	Gruppen-Trennung	ryhmän erotusmerkki
<a href="#">RS</a>	Record Separator	<a href="#">SA</a>	séparateur d'enregistrements, séparateur d'articles	<a href="#">PЗ</a>	разделитель записей	separador de registro	Untergruppen-Trennung	tietueiden erotusmerkki
<a href="#">US</a>	Unit Separator	<a href="#">SSA</a>	séparateur de sous-articles	<a href="#">PЭ</a>	разделитель элементов	separador de unidad	Teilgruppen-Trennung	yksikön erotusmerkki
<a href="#">SP</a>	Space	<a href="#">ESP</a>	espace	<a href="#">ПР</a>	пробел	espacio	Zwischenraum	tyhjä
<a href="#">DEL</a>	Delete	<a href="#">SUP</a>	suppression	<a href="#">ЗБ</a>	забой	suprimir	Löschen	merkin poisto
<a href="#">PAD</a>	"Padding Character"		caractère de bourre					
<a href="#">HOP</a>	"High Octet Preset"		octet supérieur prédéfini					
<a href="#">BPH</a>	Break Permitted Here	<a href="#">API</a>	arrêt permis ici	<a href="#">PГС</a>	разрешение переноса строки	corte permitido aquí		
<a href="#">NBH</a>	No Break Here	<a href="#">PAI</a>	aucun arrêt ici	<a href="#">ЗПС</a>	запрет переноса строки	corte no permitido aquí		
<a href="#">IND</a>	Index	<a href="#">IND</a>	index	<a href="#">ИНД</a>	индекс			
<a href="#">NEL</a>	Next Line	<a href="#">NL</a>	à la ligne	<a href="#">НС</a>	новая строка			
<a href="#">SSA</a>	Start of Selected Area	<a href="#">DZS</a>	début de zone sélectionnée	<a href="#">НВО</a>	начало выбранной области			
<a href="#">ESA</a>	End of Selected Area	<a href="#">FZS</a>	fin de zone sélectionnée	<a href="#">КВО</a>	конец выбранной области			
<a href="#">HTS</a>	Horizontal Tabulation Set	<a href="#">TTH</a>	taquet de tabulateur horizontal	<a href="#">УГТ</a>	установка горизонтальной табуляции			
<a href="#">HTJ</a>	Horizontal Tabulation with Justification	<a href="#">THJ</a>	tabulateur horizontal avec justification	<a href="#">ГТВ</a>	горизонтальная табуляция с выключкой			
<a href="#">VTS</a>	Vertical Tabulation Set	<a href="#">TTV</a>	taquet de tabulateur vertical	<a href="#">УВТ</a>	установка вертикальной табуляции			
<a href="#">PLD</a>	Partial Line Down	<a href="#">IPav</a>	interligne partiel avant	<a href="#">ССВ</a>	смещение строки вперед	avance de línea parcial		
<a href="#">PLU</a>	Partial Line Up	<a href="#">IPar</a>	interligne partiel arrière	<a href="#">ССН</a>	смещение строки назад	retroceso de línea parcial		
<a href="#">RI</a>	Reverse Index	<a href="#">IR</a>	index renversé, interligne inversé	<a href="#">ОПС</a>	обратный перевод строки	cambio de renglón inverso		
<a href="#">SS2</a>	Single Shift Two	<a href="#">RU2</a>	remplacement unique deux	<a href="#">ПЕ2</a>	переключатель единичный два	cambio individual dos		
<a href="#">SS3</a>	Single Shift Three	<a href="#">RU3</a>	remplacement unique trois	<a href="#">ПЕ3</a>	переключатель единичный три	cambio individual tres		
<a href="#">DCS</a>	Device Control String	<a href="#">CCA</a>	chaîne de commande d'appareils	<a href="#">УЦУ</a>	управляющая цепочка устройства			
<a href="#">PU1</a>	Private Use One	<a href="#">UP1</a>	usage privé un	<a href="#">ЧИ1</a>	частное использование один			
<a href="#">PU2</a>	Private Use Two	<a href="#">UP2</a>	usage privé deux	<a href="#">ЧИ2</a>	частное использование два			
<a href="#">STS</a>	Set Transmit State	<a href="#">MMT</a>	mise en mode transmission	<a href="#">УСП</a>	установка состояния передачи			

<a href="#">CCH</a>	Cancel Character	<a href="#">ANC</a>	annulation du caractère précédent	<a href="#">OTC</a>	отмена символа			
<a href="#">MW</a>	Message Waiting	<a href="#">MES</a>	message en attente	<a href="#">OC</a>	ожидание сообщения			
<a href="#">SPA</a>	Start of Guarded Protected Area	<a href="#">DZP</a>	début de zone protégée	<a href="#">HCO</a>	начало сохраняемой области			
<a href="#">EPA</a>	End of Guarded Protected Area	<a href="#">FZP</a>	fin de zone protégée	<a href="#">KCO</a>	конец сохраняемой области			
<a href="#">SOS</a>	Start of String	<a href="#">DC</a>	début de chaîne	<a href="#">HЦ</a>	начало цепочки	comienzo de cadena		
<a href="#">SGCI</a>	"Single Graphic Character Introducer"		introduceur de caractère graphique unique					
<a href="#">SCI</a>	Single Character Introducer	<a href="#">ICU</a>	introduceur de caractère unique	<a href="#">ГЕС</a>	головной символ единичного символа			
<a href="#">CSI</a>	Control Sequence Introducer	<a href="#">ISC</a>	introduceur de séquence de commandes	<a href="#">ГУП</a>	головной символ управляющей последовательности	introduccion de secuencia de control		
<a href="#">ST</a>	String Terminator	<a href="#">FC</a>	fin de chaîne	<a href="#">ТРЦ</a>	терминатор цепочки	terminador de cadena		
<a href="#">OSC</a>	Operating System Command	<a href="#">CSE</a>	commande de système d'exploitation	<a href="#">КОС</a>	команда операционной системы			
<a href="#">PM</a>	Privacy Message	<a href="#">MP</a>	message privé	<a href="#">ЧС</a>	частное сообщение			
<a href="#">APC</a>	Application Program Command	<a href="#">CO</a>	commande de logiciel	<a href="#">КПП</a>	команда прикладной программы			
<a href="#">NBSP</a>	No-Break Space	<a href="#">ESP</a>	espace insécable		непрерывающий пробел	espacio anticorte		yhdistävä välilyönti *
<a href="#">SHY</a>	Soft Hyphen	<a href="#">CDN</a>	trait d'union conditionnel		гибкий дефис	guión de corte programable		pehmeä tavuviiva *

\* Finnish terms marked with an asterisk are not from any standard, but from recommendation *Eurooppalaisen merkistön merkkien suomenkieliset nimet*.

## Character index

<a href="#">ACK</a>	Acknowledge	<a href="#">NAK</a>	Negative Acknowledge
<a href="#">APC</a>	Application Program Command	<a href="#">NBH</a>	No Break Here
<a href="#">BEL</a>	Bell	<a href="#">NBSP</a>	No-Break Space
<a href="#">BPH</a>	Break Permitted Here	<a href="#">NEL</a>	Next Line
<a href="#">BS</a>	Backspace	<a href="#">NUL</a>	Null
<a href="#">CAN</a>	Cancel	<a href="#">OSC</a>	Operating System Command
<a href="#">CCH</a>	Cancel Character	<a href="#">PAD</a>	"Padding Character" (unassigned)
<a href="#">CR</a>	Carriage Return	<a href="#">PLD</a>	Partial Line Down
<a href="#">CSI</a>	Control Sequence Introducer	<a href="#">PLU</a>	Partial Line Up
<a href="#">DC<sub>1</sub></a>	Device Control 1	<a href="#">PM</a>	Privacy Message

<a href="#"><u>DC<sub>2</sub></u></a>	Device Control 2	<a href="#"><u>PU1</u></a>	Private Use One
<a href="#"><u>DC<sub>3</sub></u></a>	Device Control 3	<a href="#"><u>PU2</u></a>	Private Use Two
<a href="#"><u>DC<sub>4</sub></u></a>	Device Control 4 (Stop)	<a href="#"><u>RI</u></a>	Reverse Index
<a href="#"><u>DCS</u></a>	Device Control String	<a href="#"><u>RS</u></a>	Record Separator
<a href="#"><u>DEL</u></a>	Delete	<a href="#"><u>SCI</u></a>	Single Character Introducer
<a href="#"><u>DLE</u></a>	Data Link Escape	<a href="#"><u>SGCI</u></a>	"Single Graphic Character Introducer" (unassigned)
<a href="#"><u>EM</u></a>	End of Medium	<a href="#"><u>SHY</u></a>	Soft Hyphen
<a href="#"><u>ENQ</u></a>	Enquiry	<a href="#"><u>SI</u></a>	Shift In
<a href="#"><u>EOT</u></a>	End of Transmission	<a href="#"><u>SO</u></a>	Shift Out
<a href="#"><u>EPA</u></a>	End of Guarded Protected Area	<a href="#"><u>SOH</u></a>	Start of Heading
<a href="#"><u>ESA</u></a>	End of Selected Area	<a href="#"><u>SOS</u></a>	Start of String
<a href="#"><u>ESC</u></a>	Escape	<a href="#"><u>SP</u></a>	Space
<a href="#"><u>ETB</u></a>	End of Transmission Block	<a href="#"><u>SPA</u></a>	Start of Guarded Protected Area
<a href="#"><u>ETX</u></a>	End of Text	<a href="#"><u>SS2</u></a>	Single Shift Two
<a href="#"><u>FE<sub>0</sub></u></a>	Format effector 0 (Backspace)	<a href="#"><u>SS3</u></a>	Single Shift Three
<a href="#"><u>FE<sub>1</sub></u></a>	Format effector 1 (Character Tabulation)	<a href="#"><u>SSA</u></a>	Start of Selected Area
<a href="#"><u>FE<sub>2</sub></u></a>	Format effector 2 (Line Feed)	<a href="#"><u>ST</u></a>	String Terminator
<a href="#"><u>FE<sub>3</sub></u></a>	Format effector 3 (Line Tabulation)	<a href="#"><u>STS</u></a>	Set Transmit State
<a href="#"><u>FE<sub>4</sub></u></a>	Format effector 4 (Form Feed)	<a href="#"><u>STX</u></a>	Start of Text
<a href="#"><u>FE<sub>5</sub></u></a>	Format effector 5 (Carriage Return)	<a href="#"><u>SUB</u></a>	Substitute
<a href="#"><u>FF</u></a>	Form Feed	<a href="#"><u>SYN</u></a>	Synchronous Idle
<a href="#"><u>FS</u></a>	File Separator	<a href="#"><u>TC<sub>1</sub></u></a>	Transmission control character 1 (Start of Heading)
<a href="#"><u>GS</u></a>	Group Separator	<a href="#"><u>TC<sub>2</sub></u></a>	Transmission control character 2 (Start of Text)
<a href="#"><u>HOP</u></a>	"High Octet Preset" (unassigned)	<a href="#"><u>TC<sub>3</sub></u></a>	Transmission control character 3 (End of Text)
<a href="#"><u>HT</u></a>	Horizontal Tabulation	<a href="#"><u>TC<sub>4</sub></u></a>	Transmission control character 4 (End of Transmission)
<a href="#"><u>HTJ</u></a>	Horizontal Tabulation with Justification	<a href="#"><u>TC<sub>5</sub></u></a>	Transmission control character 5 (Enquiry)
<a href="#"><u>HTS</u></a>	Horizontal Tabulation Set	<a href="#"><u>TC<sub>6</sub></u></a>	Transmission control character 6 (Acknowledge)
<a href="#"><u>IND</u></a>	Index	<a href="#"><u>TC<sub>7</sub></u></a>	Transmission control character 7 (Data Link Escape)
<a href="#"><u>IS<sub>1</sub></u></a>	Information separator 1 (Unit Separator)	<a href="#"><u>TC<sub>8</sub></u></a>	Transmission control character 8 (Negative Acknowledge)
<a href="#"><u>IS<sub>2</sub></u></a>	Information separator 2 (Record Separator)	<a href="#"><u>TC<sub>9</sub></u></a>	Transmission control character 9 (Synchronous Idle)
<a href="#"><u>IS<sub>3</sub></u></a>	Information separator 3 (Group Separator)	<a href="#"><u>TC<sub>10</sub></u></a>	Transmission control character 10 (End of Transmission Block)
<a href="#"><u>IS<sub>4</sub></u></a>	Information separator 4 (File Separator)	<a href="#"><u>US</u></a>	Unit Separator
<a href="#"><u>LF</u></a>	Line Feed	<a href="#"><u>VT</u></a>	Vertical Tabulation
<a href="#"><u>LS<sub>0</sub></u></a>	Locking-Shift Zero (Shift In)	<a href="#"><u>VTS</u></a>	Vertical Tabulation Set
<a href="#"><u>LS<sub>1</sub></u></a>	Locking-Shift One (Shift Out)	<a href="#"><u>XOFF</u></a>	Device Control 3
<a href="#"><u>MW</u></a>	Message Waiting	<a href="#"><u>XON</u></a>	Device Control 1

---

## Sources

- ASA standard X3.4-1963: [\*American Standard Code for Information Interchange\*](#). Note: *ASCII-1963*.
- USAS X3.4-1967: *USA Standard Code for Information Interchange*. United States of America Standards Institute, New York, USA, 1967. Note: *ASCII-1967*.
- USAS X3.4-1968: *USA Standard Code for Information Interchange*. Reprinted as NIC 11246 in Feinler & Postel (ed.): *Arpanet Protocol Handbook*. NIC 7104 Rev. Jan 1978. ADA-052 594. Network Information Center, Menlo Park, California, USA. Note: *ASCII-1968*.

- ANSI X3.4-1977: *American National Standard Code for Information Interchange*. American National Standards Institute, Inc, New York, USA, 1977. Also reprinted in McGraw Hill's *Compilation of Data Communication Standards*, edition II, McGraw-Hill, 1982. *Note: ASCII-1977.*
- ANSI X3.4-1986: *Coded Character Sets – 7-bit American National Standard Code for Information Interchange*. American National Standards Institute, Inc, New York, USA, 1986. *Note: ASCII-1986.*
- ANSI X3.32-1973: *Graphic Representation of the Control Characters of American National Standard Code for Information Interchange*. Reprinted in McGraw Hill's *Compilation of Data Communication Standards*, edition II, McGraw-Hill, 1982.
- ANSI X3.64-1979: *Additional Controls for Use with American National Standard Code for Information Interchange*. American National Standards Institute, Inc, New York, USA, 1979.
- Bemmer, R.W.: *Inside ASCII*. Best of Interface Age, Volume 2: General Purpose Software. Oregon, USA (1980).
- Bies, Lammert: [ASCII character map](#).
- Digital Research: [An Introduction to CP/M Features and Facilities](#), version 1.3, 1976.
- ECMA-6: [7-bit Coded Character Set](#), 4th edition 1973, 5th edition 1985.
- ECMA-17: [Graphic Representation of the Control Characters of the ECMA 7-Bit Coded Character Set for Information Interchange](#), 1st edition (withdrawn).
- ECMA-35: [Character Code Structure and Extension Techniques](#), 6th edition.
- ECMA-48: [Control Functions for Coded Character Sets](#), 2nd, 3rd, 4th and 5th edition.
- Gerstung, Olaf: *Tabellen – Verschiedenes*. Bedeutung der Steuerzeichen im ASCII und nach DIN 66003.
- GOST 34.301-91: [Information technology. 7-bit and 8-bit coded character sets. Control functions](#) – ГОСТ 34.301-91 (ИСО 6429-88) Информационная технология. 7-битные и 8-битные кодированные наборы символов. Управляющие функции.
- GOST 34.302.2-91: [Information technology. 8-bit single-byte coded graphic character sets. Latin alphabet No. 2](#) – ГОСТ 34.302-91 (ИСО 8859/2-87) Информационная технология. Наборы 8-битных однобайтовых кодированных графических символов. Латинский алфавит № 2.
- Helsingin yliopiston yleisen kielitieteen laitos: [Eurooppalaisen merkistön merkkien suomenkieliset nimet](#), 2. laitos, toukokuu 2004.
- ISO / R 646-1967 (E): *6 and 7-bit coded character sets for information processing interchange*, 1st edition December 1967. International Organization for Standardization, Switzerland.
- ISO 646-1973 (E): *7-bit coded character set for information processing interchange*. ISO Standards Handbook 1: Information transfer, 1st edition, 1977. Also reprinted in McGraw Hill's *Compilation of Data Communication Standards*, edition II, McGraw-Hill, 1982.
- ISO 646:1991: *Information technology – 7-bit coded character set for information processing interchange*.
- ISO 2022-1973 (E): *Code extension techniques for use with the ISO 7-bit coded character set*. ISO Standards Handbook 1: Information transfer, 1st edition, 1977.

- ISO 2047-1975 (E): *Information processing – Graphical representations for the control characters of the 7-bit coded character set*. ISO Standards Handbook 1: Information transfer, 1st edition, 1977.
- ISO/IEC 6429:1992 (E): *Information technology – Control functions for coded character sets*.
- ISO 1745-1975 (E): *Information processing – Basic mode control procedures for data communication systems*. Reprinted in McGraw Hill's Compilation of Data Communication Standards, edition II, McGraw-Hill, 1982.
- ISO/IEC 8859: *Information technology – 8-bit single-byte coded graphic character sets*. Note: Mostly ISO/IEC 8859-1:1998: *8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1*.
- ISO-IR 001: [The set of control characters of the ISO 646](#). Note: ISO-IR 001 deviates slightly from ISO 646-1973 in wording. DEL missing.
- ISO-IR 077: [C1 Control Character Set of ISO 6429-1983](#).
- Jennings, Tom: [An annotated history of some character codes](#), revised 29 October, 2004.
- RFC 20: [ASCII format for Network Interchange](#). Note: Identical to USAS X3.4-1968 (ASCII-1968). Missing Appendix A–D.
- RFC 1345: [Character Mnemonics & Character Sets](#).
- SFS 4017: *Tietojen vaihdossa käytettävä 7-bittinen koodi – 7-bit coded character set for information processing interchange*. Suomen standardisoimisliitto, Helsinki, Finland, 1977.
- UIT-T T.50 (04/92): [Alfabeto internacional de referencia](#), (anteriormente alfabeto internacional N.º 5 o IA5) – Tecnología de la información - Juego de caracteres codificado de siete bits para intercambio de información.
- UIT-T T.51 (09/92): [Juegos de caracteres codificados basados en el alfabeto latino para los servicios de telemática](#).
- UIT-T T.53 (04/94): [Funciones de control codificadas mediante caracteres para los servicios telemáticos](#).
- Unicode, Inc.: [Unicode 5.0, section française](#).
- Unicode, Inc.: [The Unicode Standard](#), version 9.0.0, 2016.
- Unicode, Inc.: [Unicode Character Database, NameAliases-9.0.0.txt](#).
- Whistler, Ken: [Why Nothing Ever Goes Away \(was: Re: Acquiring DIS 10646\)](#). Unicode Mail List, 5 Oct 2015.
- Wikipedia: [ASCII](#).
- Wikipedia: [C0 and C1 control codes](#).
- Wikipedia: [Control character](#).
- Wikipedia: [Newline](#).
- Wikipedia: [Software flow control](#).

Most of the sources have been consulted as of September/October 2011.



Special thanks for help to Douglas A. Kerr, the principal author and editor of the published standards document of the first complete version of ASCII.

Last updated in August 2016: Unicode 9.0, CP/M, additional details on PAD, HOP and SGCI.