

CRC32 & CRC64 and other checksum methods

What is CRC-64?

A **cyclic redundancy check (CRC)** is an error-detecting code used to detect data corruption. When sending data, short checksum is generated based on data content and sent along with data. When receiving data, checksum is generated again and compared with sent checksum. If the two are equal, then there is no data corruption. The **CRC-64 algorithm** itself converts a variable-length string into an 16-character string.

[crayon-61020a6a6131d860151172/]

[crayon-61020a6a61330999824098/]

[File : PDF : Click : Cyclic-Redundancy-Code-CRC-Polynomial-Selection-For-Embedded-Networks_koopman04_crc_poly_embedded](#)

File : [sha1_comparison_Which_is-a-more-accurate-method-of-duplicate-file-detection_SHA1 or CRC32_ - Quora](#)

File: CRC Data Sets Tests : [PDF-DOWNLOAD_kutayzorlu.com_18.2_million_dataset_CRC-64_test_Program_source](#)

File : Amazon Red Shift : [Pdf Download : kutayzorlu.com_amazon_red_redshift-dg](#)

[crayon-61020a6a61338131314691/]

SHA1- JAVA checksum verify

[crayon-61020a6a6133d414139187/]

SHA1- TEXT String

[crayon-61020a6a61344233561133/]

SHA1 – File – JAVA

[crayon-61020a6a6134a835059536/]

[crayon-61020a6a6134f351949322/]

References

- <https://crc64.online/>
- <http://www.unit-conversion.info/texttools/crc/>
- <https://www.nitrxgen.net/hashgen/>
- <http://www.sha1-online.com/>
- <https://toolslick.com/programming/hashing/crc-calculator#>
- https://en.wikipedia.org/wiki/Cyclic_redundancy_check
- <https://searchcode.com/codesearch/view/22078345/>
- <https://jar-download.com/artifacts/com.github.tonivade/claadb/1.0.2/source-code/com/github/tonivade/claadb/persistence/CRC64.java>
- <https://stackoverflow.com/questions/2321676/data-length-vs-crc-length>
- <https://stackoverflow.com/questions/20562546/how-to-get-crc64-distributed-calculation-use-its-linearity-property>
- <http://www.sha1-online.com/sha1-java/>