
Question

[Eduard Lebedyuk](#) · Jun 9, 2016

Cheap object hashing?

Problem: Caché instance connects periodically to an Oracle database [via jdbc](#) and downloads data. Sometimes data on a remote server would change.

Currently the loader code follows this logic:

1. Load next row
2. Check if the row exists (in Cache) or not, depending on that instantiate new object or open an existing one
3. Set object properties from the row data
4. Save row

But I don't want to reload all the data every time, so I got this idea:

1. Load next row
2. Append all row data to string with some separator and calculate CRC32 hash with [\\$zcrc](#)
3. Check if the row exists or not, if not then instantiate new object, otherwise get stored hash and compare with calculated hash. If hashes are the same then go to the next row, if not open an existing object
4. Set object properties from the row data
5. Save row

But I'm not really sure if CRC is the best solution, maybe there are some better approaches?

Maybe I can hash the row object (it's [EnsLib.SQL.Snapshot](#), but i would also be interested in a generic strategy to get a hash for object of any class)?

I don't have a write access to the Oracle database and there is no information about "last change date" to get.

[#Business Service](#) [#JDBC](#) [#Object Data Model](#) [#ODBC](#) [#Caché](#)

Source URL: <https://community.intersystems.com/post/cheap-object-hashing>