

This directory contains all Redis dependencies, except for the libc that should be provided by the operating system.

- **Jemalloc** is our memory allocator, used as replacement for libc malloc on Linux by default. It has good performances and excellent fragmentation behavior. This component is upgraded from time to time.
- **hiredis** is the official C client library for Redis. It is used by redis-cli, redis-benchmark and Redis Sentinel. It is part of the Redis official ecosystem but is developed externally from the Redis repository, so we just upgrade it as needed.
- **linennoise** is a readline replacement. It is developed by the same authors of Redis but is managed as a separated project and updated as needed.
- **lua** is Lua 5.1 with minor changes for security and additional libraries.

How to upgrade the above dependencies

Jemalloc

Jemalloc is modified with changes that allow us to implement the Redis active defragmentation logic. However this feature of Redis is not mandatory and Redis is able to understand if the Jemalloc version it is compiled against supports such Redis-specific modifications. So in theory, if you are not interested in the active defragmentation, you can replace Jemalloc just following these steps:

1. Remove the jemalloc directory.
2. Substitute it with the new jemalloc source tree.
3. Edit the Makefile located in the same directory as the README you are reading, and change the `--with-version` in the Jemalloc configure script options with the version you are using. This is required because otherwise Jemalloc configuration script is broken and will not work nested in another git repository.

However note that we change Jemalloc settings via the `configure` script of Jemalloc using the `--with-lg-quantum` option, setting it to the value of 3 instead of 4. This provides us with more size classes that better suit the Redis data structures, in order to gain memory efficiency.

If you want to upgrade Jemalloc while also providing support for active defragmentation, in addition to the above steps you need to perform the following additional steps:

5. In Jemalloc tree, file `include/jemalloc/jemalloc_macros.h.in`, make sure to add `#define JEMALLOC_FRAG_HINT`.
6. Implement the function `je_get_defrag_hint()` inside `src/jemalloc.c`. You can see how it is implemented in the current Jemalloc source tree shipped with Redis, and rewrite it according to the new Jemalloc internals, if they changed, otherwise you could just copy the old implementation if you are upgrading just to a similar version of Jemalloc.

Updating/upgrading jemalloc

The jemalloc directory is pulled as a subtree from the upstream jemalloc github repo. To update it you should run from the project root:

1.

```
git subtree pull --prefix deps/jemalloc https://github.com/jemalloc/jemalloc.git <version-tag> --squash
```

This should hopefully merge the local changes into the new version.
2. In case any conflicts arise (due to our changes) you'll need to resolve them and commit.
3. Reconfigure jemalloc:

```
rm deps/jemalloc/VERSION deps/jemalloc/configure
cd deps/jemalloc
./autogen.sh --with-version=<version-tag>-0-g0
```

4. Update jemalloc's version in `deps/Makefile` : search for " `--with-version=<old-version-tag>-0-g0` " and update it accordingly.
5. Commit the changes (VERSION,configure,Makefile).

Hiredis

Hiredis uses the SDS string library, that must be the same version used inside Redis itself. Hiredis is also very critical for Sentinel. Historically Redis often used forked versions of hiredis in a way or the other. In order to upgrade it is advised to take a lot of care:

1. Check with diff if hiredis API changed and what impact it could have in Redis.
2. Make sure that the SDS library inside Hiredis and inside Redis are compatible.
3. After the upgrade, run the Redis Sentinel test.
4. Check manually that redis-cli and redis-benchmark behave as expected, since we have no tests for CLI utilities currently.

Linenoise

Linenoise is rarely upgraded as needed. The upgrade process is trivial since Redis uses a non modified version of linenoise, so to upgrade just do the following:

1. Remove the linenoise directory.
2. Substitute it with the new linenoise source tree.

Lua

We use Lua 5.1 and no upgrade is planned currently, since we don't want to break Lua scripts for new Lua features: in the context of Redis Lua scripts the capabilities of 5.1 are usually more than enough, the release is rock solid, and we definitely don't want to break old scripts.

So upgrading of Lua is up to the Redis project maintainers and should be a manual procedure performed by taking a diff between the different versions.

Currently we have at least the following differences between official Lua 5.1 and our version:

1. Makefile is modified to allow a different compiler than GCC.
2. We have the implementation source code, and directly link to the following external libraries:
`lua_cjson.o` , `lua_struct.o` , `lua_cmsgpack.o` and `lua_bit.o` .
3. There is a security fix in `ldo.c` , line 498: The check for `LUA_SIGNATURE[0]` is removed in order to avoid direct bytecode execution.