Node.js startup snapshot builder

This is the V8 startup snapshot builder of Node.js. Not to be confused with V8's own snapshot builder, which builds a snapshot containing JavaScript builtins, this builds a snapshot containing Node.js builtins that can be deserialized on top of V8's own startup snapshot. When Node.js is launched, instead of executing code to bootstrap, it can deserialize the context from an embedded snapshot, which readily contains the result of the bootstrap, so that Node.js can start up faster.

Currently only the main context of the main Node.js instance supports snapshot deserialization, and the snapshot does not yet cover the entire bootstrap process. Work is being done to expand the support.

How it's built and used

The snapshot builder is built with the `node_mksnapshot` target in `node.gyp` when `node_use_node_snapshot` is set to true, which is currently done by default.

In the default build of the Node.js executable, to embed a V8 startup snapshot into the Node.js executable, `libnode` is first built with these unresolved symbols:

- `node::NodeMainInstance::GetEmbeddedSnapshotData`

Then the `node_mksnapshot` executable is built with C++ files in this directory, as well as `src/node_snapshot_stub.cc` which defines the unresolved symbols.

`node_mksnapshot` is run to generate a C++ file `<(SHARED_INTERMEDIATE_DIR)/node_snapshot.cc` that is similar to `src/node_snapshot_stub.cc` in structure, but contains the snapshot data written as static char array literals. Then `libnode` is built with `node_snapshot.cc` to produce the final Node.js executable with the snapshot data embedded.

For debugging, Node.js can be built without Node.js's own snapshot if `--without-node-snapshot` is passed to `configure`. A Node.js executable with Node.js snapshot embedded can also be launched without deserializing from it if the command line argument `--no-node-snapshot` is passed.