spatial cheat sheet

- Unless otherwise specified, run commands from the root of your SpatialOS project.
- For more verbose output on any command, use `--log_level=debug`.

Setup

`spatial setup install-dependencies [--sdk-version=<version>]`

(Windows only) Install everything needed to develop with SpatialOS for your project's SDK version.

If outside a SpatialOS project directory, specify your SDK version.

`spatial diagnose`

Check if everything needed to develop with SpatialOS is installed.

When requesting help on the forums, include the output of this command.

`spatial worker build`

Run this when you use a project for the first time.

`spatial version`

Display the version of `spatial` and the SDK versions you have downloaded.

`spatial update`

Update `spatial` to the latest version.

Build workers

`spatial worker clean`

Clean worker directories.

`spatial worker codegen [worker type(s)]`

Generate code for workers from the schema.

To apply to a specific worker, run from within its directory, or specify the worker name as an argument.

To apply to all workers, run from the project root with no arguments.

`spatial worker build [worker type(s)] [--target=<OS>]`

Build workers.

To speed up build times:
- specify which worker(s) to build (eg `spatial worker build MyCSharpClient`)
- specify an operating system to build for (eg `--target=Windows`)

Deploy locally

`spatial local launch [launch config] [--snapshot=<snapshot file>]`

Start a local deployment using the launch configuration you specify.

If you don't specify a launch configuration, it defaults to "default_launch.json" in the root of your project.

`spatial local worker launch <worker type> <launch config> [--<launch config args>]`

Launch a local worker to connect to your local deployment.

For example, if your worker.json is "spatialos.MyCSharpClient.worker.json", run `spatial local worker launch MyCSharpClient default`.

Deploy to the cloud

`spatial cloud upload <assembly name>`

Upload an assembly to use for cloud deployments.

`spatial cloud launch <assembly name> <launch config> <deployment name>`

Launch a cloud deployment.

`spatial cloud connect external <deployment name>`

Set up a proxy so you can connect a worker on your machine to a cloud deployment.

`spatial cloud delete <deployment name>`

Stop a deployment.

`spatial cloud history snapshot create <deployment name>`

Take a snapshot of a deployment.

`spatial cloud history snapshot list <deployment name> [--from yyyy-mm-dd --until yyyy-mm-dd]`

List snapshots for a project within a time range (defaults to current day).

SpatialOS projects

`spatial project history clone <existing deployment> <new deployment>`

Clone a deployment's tags and snapshot set to create a new deployment.

`spatial project history snapshot convert --input=<binary.snapshot> --output=<text.txt> --input-format=binary --output-format=text`

Convert a binary snapshot file to a text snapshot file, or vice versa.

You must have uploaded your assembly since the last schema change.

Manage schemas

`spatial schema check`

Check if the schema contains errors.

`spatial schema print`

Print all components defined in schema and dependencies, and their component IDs.