

The `rustc-std-workspace-core` crate

This crate is a shim and empty crate which simply depends on `libcore` and reexports all of its contents. The crate is the crux of empowering the standard library to depend on crates from crates.io

Crates on crates.io that the standard library depend on need to depend on the `rustc-std-workspace-core` crate from crates.io, which is empty. We use `[patch]` to override it to this crate in this repository. As a result, crates on crates.io will draw a dependency edge to `libcore`, the version defined in this repository. That should draw all the dependency edges to ensure Cargo builds crates successfully!

Note that crates on crates.io need to depend on this crate with the name `core` for everything to work correctly. To do that they can use:

```
core = { version = "1.0.0", optional = true, package = 'rustc-std-workspace-core' }
```

Through the use of the `package` key the crate is renamed to `core`, meaning it'll look like

```
--extern core=.../librustc_std_workspace_core-XXXXXXX.rlib
```

when Cargo invokes the compiler, satisfying the implicit `extern crate core` directive injected by the compiler.