Treating Build Definitions Independent of their Origin

Klaus Aehlig

February 2025

Makefile

```
all: app

foo.o: foo.cpp foo.hpp
	c++ -c foo.cpp -o foo.o

libfoo.a: foo.o
	ar cqs libfoo.a foo.o

bar.o: bar.cpp bar.hpp foo.hpp
	c++ -c bar.cpp -o bar.o

libbar.a: bar.o
	ar cqs libbar.a bar.o
```

```
main.o: main.cpp bar.hpp foo.hpp
	c++ -c main.cpp -o main.o

app: main.o libbar.a libfoo.a
	c++ -o app main.o libbar.a libfoo.a

clean:
	rm -f *.o *.a app

.PHONY: all clean
```

- `Makefile`
  - files referred to by path where they will be created
  - actions for their side effect
  - `.PHONY` targets purely for their side effect
- Now imagine, we instead focused on the actual definitions ...

The actually-defined objects

`foo.o` can be defined as the output of a particular action ...

```
{
  "foo.o": {
    "data": {
      "id": "911556ffed2a47c99a03394427915fd94c8860618af7eab9de233824af1c8b5a",
      "path": "foo.o"
    },
    "type": "ACTION"
  }
}
```

The actually-defined objects

That action is defined as


```
{
  "command": [
    "sh",
    "-c",
    "c++ -c foo.cpp -o foo.o\n"
  ],
  "env": {
    "PATH": "/bin:/usr/bin"
  },
  "input": {
    "foo.cpp": {
      "data": {
        "file_type": "f",
        "id": "223e5e3701f42fe5584a...
        "size": 111
      },
      "type": "KNOWN"
    },
    "foo.hpp": {
      "data": {
        "file_type": "f",
        "id": "6bf31ad480f47c568beb...
        "size": 53
      },
      "type": "KNOWN"
    }
  },
  "output": [
    "foo.o"
  ]
}
```

The actually-defined objects

`libfoo.a` can be defined as the output of a particular action ...

```
{
  "libfoo.a": {
    "data": {
      "id": "018f7966fd9e6aef14d26788e23edd077dbfa966035854998343e0aef79579bd",
      "path": "libfoo.a"
    },
    "type": "ACTION"
  }
}
```

The actually-defined objects

That action is defined as

```
{
  "command": [
    "sh",
    "-c",
    "ar cqs libfoo.a foo.o\n"
  ],
  "env": {
    "PATH": "/bin:/usr/bin"
  },
  "input": {
    "foo.o": {
      "data": {
        "id": "911556ffed2a47c99a03...
        "path": "foo.o"
      },
      "type": "ACTION"
    }
  },
  "output": [
    "libfoo.a"
  ]
}
```

Artifact Definitions

- Artifacts can be defined as ...
  - source files, named, e. g., by its `git` blob identifier
  - particular output of an action given by
    - mapping of logical paths to input-artifact definition
    - argument vector
    - environment variables

    named by some appropriate hash thereof.

⇝ Action graph

- actions defined independent of the place they're executed
  (can even be sent to a remote-execution service)
- action evaluation can also be cached independently of action provenance

- Files are given by their definition, not a place

Targets

Again, the same `Makefile`

```
all: app

foo.o: foo.cpp foo.hpp
	c++ -c foo.cpp -o foo.o

libfoo.a: foo.o
	ar cqs libfoo.a foo.o

bar.o: bar.cpp bar.hpp foo.hpp
	c++ -c bar.cpp -o bar.o

libbar.a: bar.o
	ar cqs libbar.a bar.o
```

```
main.o: main.cpp bar.hpp foo.hpp
	c++ -c main.cpp -o main.o

app: main.o libbar.a libfoo.a
	c++ -o app main.o libbar.a libfoo.a

clean:
	rm -f *.o *.a app

.PHONY: all clean
```

- `Makefile`, implicit structure
  - libraries `foo` and `bar`, and binary `app`
  - dependencies: `bar` publicly depends on `foo` (see propagation of header `foo.hpp`)
- Let's make that structure explicit ...

Targets

A more structured description of the same

```
{ "foo":                                          | , "app":
  { "type": ["@", "rules", "CC", "library"]       |   { "type": ["@", "rules", "CC", "binary"]
  , "name": ["foo"]                               |   , "name": ["app"]
  , "hdrs": ["foo.hpp"]                           |   , "srcs": ["main.cpp"]
  , "srcs": ["foo.cpp"]                           |   , "private-deps": ["bar"]
  }                                               |   }
, "bar":                                          | }
  { "type": ["@", "rules", "CC", "library"]       |
  , "name": ["bar"]                               |
  , "hdrs": ["bar.hpp"]                           |
  , "srcs": ["bar.cpp"]                           |
  , "deps": ["foo"]                               |
  }                                               |
```

Analysed Targets

Result of analysing `"foo"`


```
{
  "artifacts": {
    "libfoo.a": {
      "data": {
        "id": "be8956137a1fc8938...
        "path": "work/libfoo.a"
      },
      "type": "ACTION"
    }
  },
  "provides": {
    "compile-args": [],
    "compile-deps": {},
    "debug-hdrs": {},
    "debug-srcs": {},
    "link-args": [
      "libfoo.a"
    ],
    "link-deps": {},
    "lint": [],
    "package": {
      "cflags-files": {},
      "ldflags-files": {},
      "name": "foo"
    }
  },
  "runfiles": {
    "foo.hpp": {
      "data": {
        "file_type": "f",
        "id": "6bf31ad480f47c568...
        "size": 53
      },
      "type": "KNOWN"
    }
  }
}
```

Analysed Targets

Result of analysing "bar"


```
{
  "artifacts": {
    "libbar.a": {
      "data": {
        "id": "aac7a11cc78afefb0...
        "path": "work/libbar.a"
      },
      "type": "ACTION"
    }
  },
  "provides": {
    "compile-args": [],
    "compile-deps": {
      "foo.hpp": {
        "data": {
          "file_type": "f",
          "id": "6bf31ad480f47c5...
          "size": 53
        },
        "type": "KNOWN"
      }
    },
    "debug-hdrs": {},
    "debug-srcs": {},
    "link-args": [
      "libbar.a",
      "libfoo.a"
    ],
    "link-deps": {
      "libfoo.a": {
        "data": {
          "id": "be8956137a1fc89...
          "path": "work/libfoo.a"
        },
        "type": "ACTION"
      }
    },
    "lint": [],
    "package": {
      "cflags-files": {},
      "ldflags-files": {},
      "name": "bar"
    }
  },
  "runfiles": {
    "bar.hpp": {
      "data": {
        "file_type": "f",
        "id": "427bf71c95adb1f0d...
        "size": 53
      },
      "type": "KNOWN"
    }
  }
}
```

Evaluated Targets

Evaluated result of analysing "bar"


```
{
  "artifacts": {
    "libbar.a": {
      "data": {
        "file_type": "f",
        "id": "b94c10968fefa3a1b...
        "size": 2816
      },
      "type": "KNOWN"
    }
  },
  "provides": {
    "compile-args": [],
    "compile-deps": {
      "foo.hpp": {
        "data": {
          "file_type": "f",
          "id": "6bf31ad480f47c5...
          "size": 53
        },
        "type": "KNOWN"
      }
    },
    "debug-hdrs": {},
    "debug-srcs": {},
    "link-args": [
      "libbar.a",
      "libfoo.a"
    ],
    "link-deps": {
      "libfoo.a": {
        "data": {
          "file_type": "f",
          "id": "6e4334d2748e6a9...
          "size": 2760
        },
        "type": "KNOWN"
      }
    },
    "lint": [],
    "package": {
      "cflags-files": {},
      "ldflags-files": {},
      "name": "bar"
    }
  },
  "runfiles": {
    "bar.hpp": {
      "data": {
        "file_type": "f",
        "id": "427bf71c95adb1f0d...
        "size": 53
      },
      "type": "KNOWN"
    }
  }
}
```

Targets

- Define logical units (library, binary)
- Defined by reference to
  - language-specific rules
  - other targets
- Evaluating the rules gives intensional description
  ... and this is everything other targets need—and can see!
- Evaluating the artifacts in that description gives essentially a package
- Again, everything independent of provenance

Modified Targets File

Target description so far

```
{ "foo":                                                  | , "app":
  { "type": ["@", "rules", "CC", "library"]               |   { "type": ["@", "rules", "CC", "binary"]
  , "name": ["foo"]                                       |   , "name": ["app"]
  , "hdrs": ["foo.hpp"]                                   |   , "srcs": ["main.cpp"]
  , "srcs": ["foo.cpp"]                                   |   , "private-deps": ["bar"]
  }                                                       |   }
, "bar":                                                  | }
  { "type": ["@", "rules", "CC", "library"]               |
  , "name": ["bar"]                                       |
  , "hdrs": ["bar.hpp"]                                   |
  , "srcs": ["bar.cpp"]                                   |
  , "deps": ["foo"]                                       |
  }                                                       |
```

Modified Targets File

Target description with `"bar"` marked as export

```
{ "foo":
  { "type": ["@", "rules", "CC", "library"]
  , "name": ["foo"]
  , "hdrs": ["foo.hpp"]
  , "srcs": ["foo.cpp"]
  }
, "bar (unexported)":
  { "type": ["@", "rules", "CC", "library"]
  , "name": ["bar"]
  , "hdrs": ["bar.hpp"]
  , "srcs": ["bar.cpp"]
  , "deps": ["foo"]
  }
```

```
| , "bar":
|   { "type": "export"
|   , "target": "bar (unexported)"
|   , "flexible_config": ["TOOLCHAIN_CONFIG", "DEBUG"]
|   }
| , "app":
|   { "type": ["@", "rules", "CC", "binary"]
|   , "name": ["app"]
|   , "srcs": ["main.cpp"]
|   , "private-deps": ["bar"]
|   }
| }
|
```

Repository Description

Repository configuration (human written)

```
{ "repositories":
  { "":
    { "repository": {"type": "file", "path": ".", "pragma": {"to_git": true}}
    , "bindings": {"rules": "rules"}
    }
  , "rules":
    { "repository":
      {"type": "file", "path": "../rules", "pragma": {"to_git": true}}
    }
  }
}
```

Repository Description

Repository configuration (derived)


```
{
  "repositories": {
    "": {
      "bindings": {
        "rules": "rules"
      },
      "workspace_root": [
        "git tree",
        "f29bfbe990b2cedb127bf71e02c92820be980cd7",
        "/worker/build/6241ca63b018c227/root/work/src"
      ]
    },
    "rules": {
      "workspace_root": [
        "git tree",
        "015465c4fef2481fc7ae91b288a00da1bdd721bf",
        "/worker/build/6241ca63b018c227/root/work/rules"
      ]
    }
  }
}
```

Building

Analysing "bar"

```
$ just-mr analyse bar --log-limit 4
INFO: Performing repositories setup
INFO: Found 2 repositories to set up
INFO: Setup finished, exec ["just","analyse","-C","/worker/build/6241ca63b018c227/root/home/.cache/just/protocol-depe...
INFO: Requested target is [["@","","","bar"],{}]
PERF: Export target ["@","","","bar"] registered for caching: [7c388143a19897c999e666c2064b3eef4613ba1b:163:f]
INFO: Analysed target [["@","","","bar"],{}]
INFO: Export targets found: 0 cached, 1 uncached, 0 not eligible for caching
INFO: Result of target [["@","","","bar"],{}]: {
        "artifacts": {
          "libbar.a": {"data":{"id":"aac7a11cc78afefb0c8df28ed6bb2904ad9381511814c2ebb4ef2b6b90991e5c","path":"work/l...
        },
        "provides": {
          "compile-args": [
          ],
          "compile-deps": {
            "foo.hpp": {"data":{"file_type":"f","id":"6bf31ad480f47c568bebcb638f3fd3e037ca5870","size":53},"type":"KN...
          },
          "debug-hdrs": {
          },
          "debug-srcs": {
          },
          "link-args": [
```

Building

Cache key


```
{
  "effective_config": "{\"DEBUG\":null,\"TOOLCHAIN_CONFIG\":null}",
  "repo_key": "5054a19f70a01ed13113cc1a300335a882004acc",
  "target_name": "[\"\",\"bar\"]"
}
```

Building

Repository description referenced by the key


```
{
  "0": {
    "bindings": {
      "rules": "1"
    },
    "expression_file_name": "EXP...
    "expression_root": [
      "git tree",
      "f29bfbe990b2cedb127bf71e0...
    ],
    "rule_file_name": "RULES",
    "rule_root": [
      "git tree",
      "f29bfbe990b2cedb127bf71e0...
    ],
    "target_file_name": "TARGETS",
    "target_root": [
      "git tree",
      "f29bfbe990b2cedb127bf71e0...
    ],
    "workspace_root": [
      "git tree",
      "f29bfbe990b2cedb127bf71e0...
    ]
  },
  "1": {
    "bindings": {},
    "expression_file_name": "EXP...
    "expression_root": [
      "git tree",
      "015465c4fef2481fc7ae91b28...
    ],
    "rule_file_name": "RULES",
    "rule_root": [
      "git tree",
      "015465c4fef2481fc7ae91b28...
    ],
    "target_file_name": "TARGETS",
    "target_root": [
      "git tree",
      "015465c4fef2481fc7ae91b28...
    ],
    "workspace_root": [
      "git tree",
      "015465c4fef2481fc7ae91b28...
    ]
  }
}
```

Building

Building something depending on "bar"

```
$ just-mr build
INFO: Performing repositories setup
INFO: Found 2 repositories to set up
INFO: Setup finished, exec ["just","build","-C","/worker/build/6241ca63b018c227/root/home/.cache/just/protocol-depend...
INFO: Requested target is [["@","","","app"],{}]
INFO: Analysed target [["@","","","app"],{}]
INFO: Export targets found: 0 cached, 1 uncached, 0 not eligible for caching
INFO: Discovered 6 actions, 3 trees, 0 blobs
INFO: Building [["@","","","app"],{}].
INFO: Processed 6 actions, 0 cache hits.
INFO: Artifacts built, logical paths are:
        app [010d6b33c07530ac52f511879404b614201e7634:17328:x]
INFO: Backing up artifacts of 1 export targets
$
```

Building

Analysing "bar", again

```
$ just-mr analyse bar --log-limit 4
INFO: Performing repositories setup
INFO: Found 2 repositories to set up
INFO: Setup finished, exec ["just","analyse","-C","/worker/build/6241ca63b018c227/root/home/.cache/just/protocol-depe...
INFO: Requested target is [["@","","","bar"],{}]
PERF: Export target ["@","","","bar"] taken from cache: [7c388143a19897c999e666c2064b3eef4613ba1b:163:f] -> [c3885179...
INFO: Analysed target [["@","","","bar"],{}]
INFO: Export targets found: 1 cached, 0 uncached, 0 not eligible for caching
INFO: Result of target [["@","","","bar"],{}]: {
        "artifacts": {
          "libbar.a": {"data":{"file_type":"f","id":"b94c10968fefa3a1b19e2958b0b3834ab24370b4","size":2816},"type":"K...
        },
        "provides": {
          "compile-args": [
          ],
          "compile-deps": {
            "foo.hpp": {"data":{"file_type":"f","id":"6bf31ad480f47c568bebcb638f3fd3e037ca5870","size":53},"type":"KN...
          },
          "debug-hdrs": {
          },
          "debug-srcs": {
          },
          "link-args": [
```

Building

Now there is nothing to build for `"bar"`

```
$ just-mr build bar -s
INFO: Performing repositories setup
INFO: Found 2 repositories to set up
INFO: Setup finished, exec ["just","build","-C","/worker/build/6241ca63b018c227/root/home/.cache/just/protocol-depend...
INFO: Requested target is [["@","","","bar"],{}]
INFO: Analysed target [["@","","","bar"],{}]
INFO: Export targets found: 1 cached, 0 uncached, 0 not eligible for caching
INFO: Discovered 0 actions, 0 trees, 0 blobs
INFO: Building [["@","","","bar"],{}].
INFO: Processed 0 actions, 0 cache hits.
INFO: Artifacts built, logical paths are:
        libbar.a [b94c10968fefa3a1b19e2958b0b3834ab24370b4:2816:f]
        bar.hpp [427bf71c95adb1f0db3c0bb9c6044fbc594528d4:53:f]
$
```

One more thing . . .

- Tree identifier is a function of the typical repository-root descriptions: commit id, the result of unpacking an archive with a given blob id, . . .

⇝ Can provide this mapping as a service
(Of course, that service needs all the sources; but they need to be stored anyway)

- From only that can compute the cache key for a target

⇝ Can provide a service evaluating and caching the target
. . . and uploading the artifacts to the remote-execution service

- So, as a user, can simply checkout only the repo to work on and dependencies are taken care of—even in a bootstrapped setup!
- Additonal advantage: different cache rotation frequencies

tl;dr

Summary

- Definitions are data structures in their own right.
- Focus on *how* things are defined, not *where*.

⇝ Seemless transition between building and package building

All free and open-source software!

- `https://github.com/just-buildsystem/justbuild`
  License: Apache-2.0
- Packaged in `AUR`, `Debian`, `nixpkgs`, `spack`, `Void`