

DataDash Specification (v1.0)

File Extension: .dash

Purpose: Human-friendly, flexible, extended alternative to JSON.

1. Core Goals

DataDash is designed to be:

- Human-readable
- Faster to write than JSON
- Strict enough to avoid ambiguity
- Flexible enough to replace JSON, YAML, and TOML in configs
- Easily parseable

2. Basic Structure

A .dash file is composed of:

- Key-value pairs
- Lists
- Primitives (strings, numbers, booleans, null)
- Comments
- Optional extended types

3. Data Types

Strings:

- Quoted: "Hello"
- Multiline: """ text """

Numbers:

- Integer, Float, Hex, Binary, Infinity, NaN

Booleans:

true, false

Null:

null

Lists:

```
colors: [  
  "red",  
  "green",  
]
```

Objects:

```
window: {  
  width: 1280,  
  height: 720,  
}
```

4. Keys

Unquoted keys allowed unless containing spaces.

5. Comments

```
// C-style  
# inline Python style
```

6. Trailing Commas

Allowed in lists and objects.

7. Extended Types

```
created: @2025-11-19T18:00:00Z  
id: @uuid("ce29...")  
icon: !base64("data")
```

8. Anchors & References

```
shared: &cfg; { lives: 3 }  
clone: *cfg
```

9. Expressions

```
area: =width * height
```

10. Whitespace

Ignored except inside strings.

11. Root Structure

Root can be an object or list.

12. Errors

Parsers must provide readable errors.

13. Compatibility

Easy conversion to/from JSON.

14. Full Example

```
app: {  
  name: "Ultra Player",
```

```
version: "2.5",
debug: true,
window: {
width: 800,
height: 600,
fullscreen: false,
},
colors: [
"red",
"green",
"blue",
],
created: @2025-11-19T10:22:00Z,
description: """
Multi-line text.

""",
shared: &cfg; {
lives: 3,
ammo: 90,
},
clone: *cfg,
}
```