DYALOG

Elsinore 2019

Cross-platform configuration files

Richard Smith

What will they be?

- Files containing configuration settings for the Dyalog environment that are:
- Portable across different platforms, and
- Human readable and writable, and
- Are simple to distribute, and
- Can be applied per-application, and
- Can be cascaded

Why have another way of setting options?

- Currently, options are easy to set locally for all APL apps. For anything else:
  - You have to set up scripts and/or command-line options and/or mess with the registry
  - You have to do this differently on different platforms
- Configuration files will be easier to use and consistent across all platforms

When will they be available?

- Next year
- Currently in early stages of development

Setting configuration options today

What the ~~Windows~~ UNIX documentation says …

You can change the parameters in 4 ways:

- ~~Using the Configuration dialog box that is obtained by selecting Configure from the Options menu on the Dyalog APL/W session.~~
- ~~By directly editing the Windows Registry using REGEDIT.EXE or REGEDIT32.EXE.~~ ~/.Dyalog/Dyalog.config
- By defining the parameters as environment variables. in upper case
- By defining the parameters on the APL command line.

This scheme provides a greater deal of flexibility, and a system whereby you can override one setting with another. For example, you can define your normal workspace size (maxws) in the Registry, but override it with a new value specified on the APL command line.

Cross-platform configuration files

Configuration files are loaded implicitly…

- If found in the current working directory

```
%USERPROFILE%/Dyalog APL-64 18.0
Unicode Files/dyalog_config
```

- If found in the same place as a workspace specified at startup

```
dyalog /path/to/workspace.dws
```

```
/path/to/dyalog.config
```

- (Whichever is found first)

Configuration files will also be loaded …

- If given as a startup file

```
dyalog /path/to/my_application
```

```
/path/to/my_application
```

- Config file then "chains" the actual startup file.

Cross-platform configuration files

- Will be supported in addition to the existing methods
  - No *requirement* to use configuration files

Precedence

Registry *or* ~/.dyalog/dyalog.config

Implicitly found configuration file

Environment variables

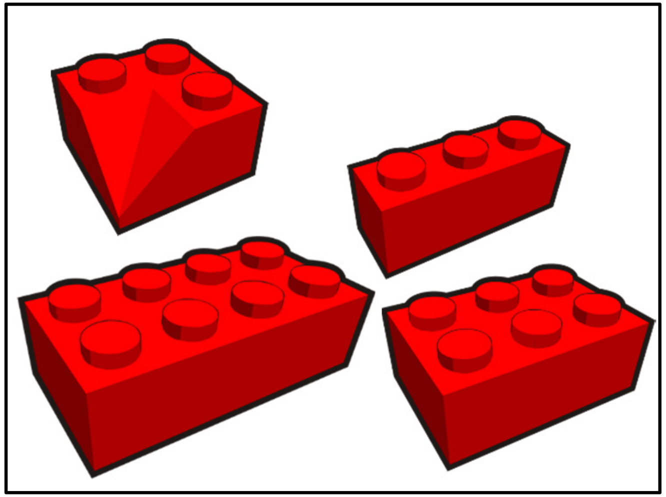
Interpreter command line

Explicitly specified configuration file

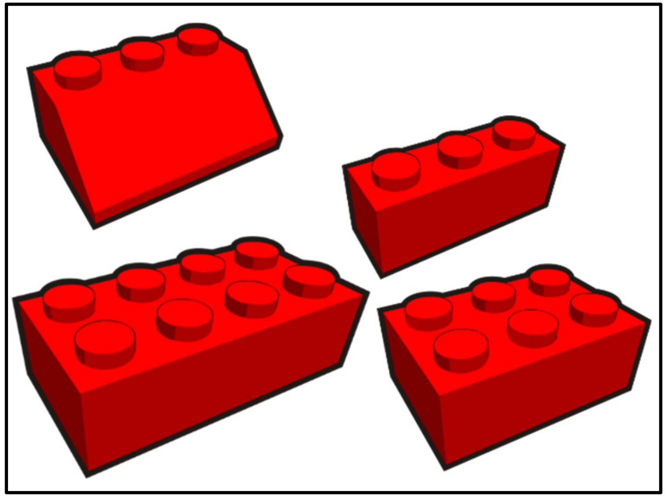
Lower priority

Higher priority

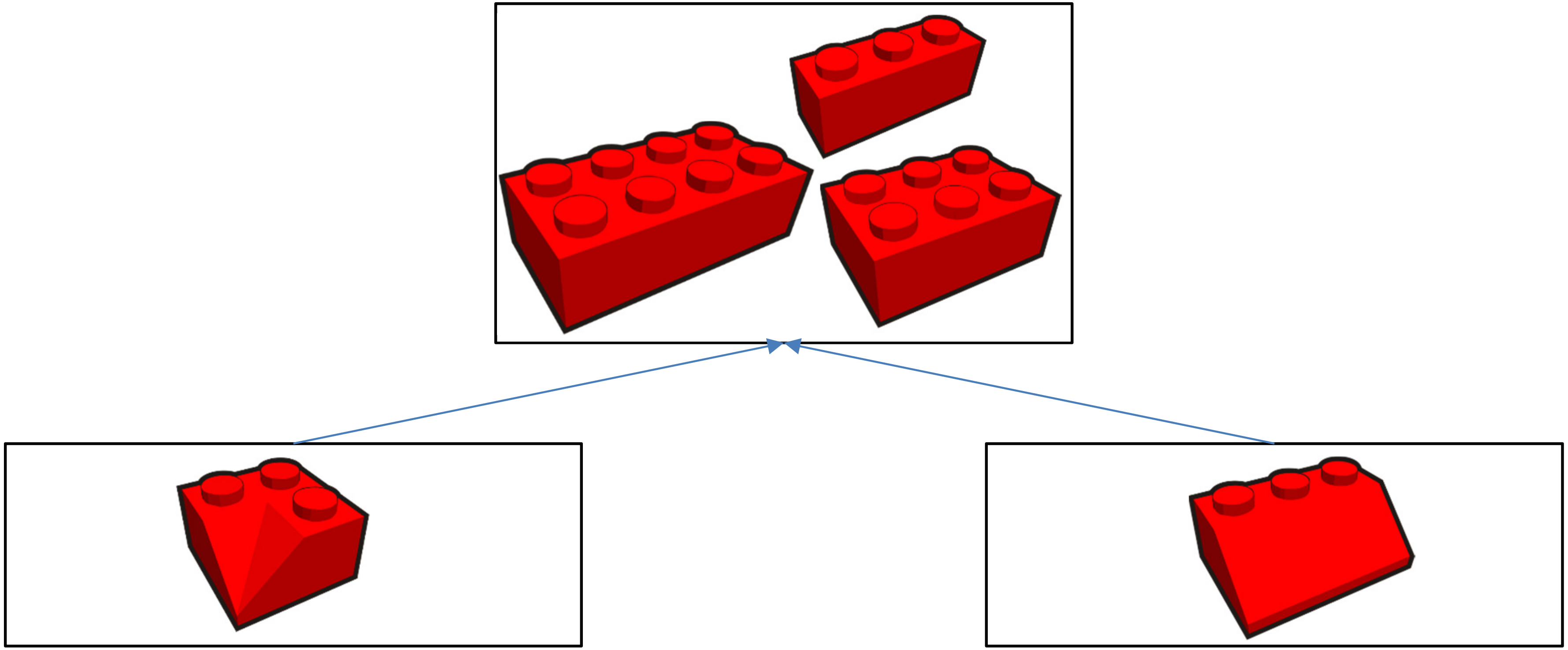
Managing separate configurations

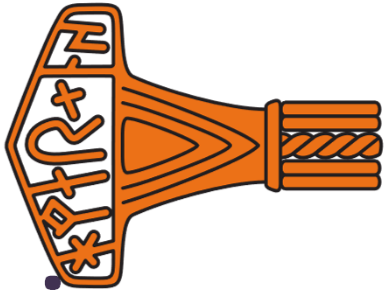


Lots of duplication



Managing separate configurations





Lots of things are still to be worked out

File format

- JSON

  …probably

Advantages of JSON

- Is human-readable and easily edited
- Can describe both character and numeric data
- Can already be read/written by the interpreter
- Can be read/written by an application using `⎕NGET`/`⎕NPUT` and `⎕JSON`.

Example JSON configuration file

```
{
        "AddClassHeaders": 0,
        "APL_FCREATE_PROPS_J": 3,
        "CFEXT": ".dcf:.DCF:",
        "confirm_fix": 0,
        "MAXWS": "1G"
}
```

JSON issues

- Does not support cascading

Possible solution

```
{
    "AddClassHeaders": 0,
    "Inherit": "common_options_file"
}
```

JSON issues

```
{
        "AddClassHeaders": 0,
        "APL_FCREATE_PROPS_J": 3,
        "CFEXT": ".dcf:.DCF:"
}
```

Possible solution

```
{
        AddClassHeaders: 0,
        APL_FCREATE_PROPS_J: 3,
        CFEXT: ".dcf:.DCF:"
}
```

JSON issues

```
{
    MAXWS: "1G",
    APL_FCREATE_PROPS_J: 3,
    MAXWS: "2G"
}
```

Possible solution

```
{
      MAXWS: "1G",
      APL_FCREATE_PROPS_J: 3,
      MAXWS: "2G"      ✖
}
```

JSON issues

- Does not support comments

Possible solution - 1

```
{
    // Maximum workspace usage
    MAXWS: "1G"
}
```

Possible solution - 2

```
{
    MAXWS: {
        Desc: "Maximum workspace usage",
        Value: "1G"
    }
}
```

~~CASE SENSITIVITY~~ Case sensitivity

- The Windows documentation lists config names in various case combinations
- On Windows they may be specified in any case
- On non-Windows they must be specified in ALL-CAPS

```
AddClassHeaders
APL_FCREATE_PROPS_J
CFEXT
maxws
```

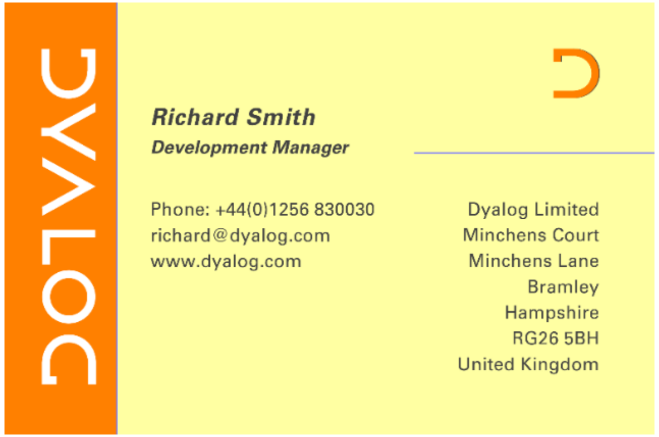
~~CASE SENSITIVITY~~ Case sensitivity

- Config file could:
  - Allow the name to given in any case, or
  - Require that the name be ALL-CAPS, or
  - Require that the name be exactly as documented
- But it must be the same on all platforms

Summary

- Simpler setting of options, particularly
  - Per application
  - When distributing an application
- Portable
- Available in next release

- Questions?



DYALOG

Richard Smith
Development Manager

Phone: +44(0)1256 830030
richard@dyalog.com
www.dyalog.com

Dyalog Limited
Minchens Court
Minchens Lane
Bramley
Hampshire
RG26 5BH
United Kingdom

support@dyalog.com

richard@dyalog.com

forums.dyalog.com