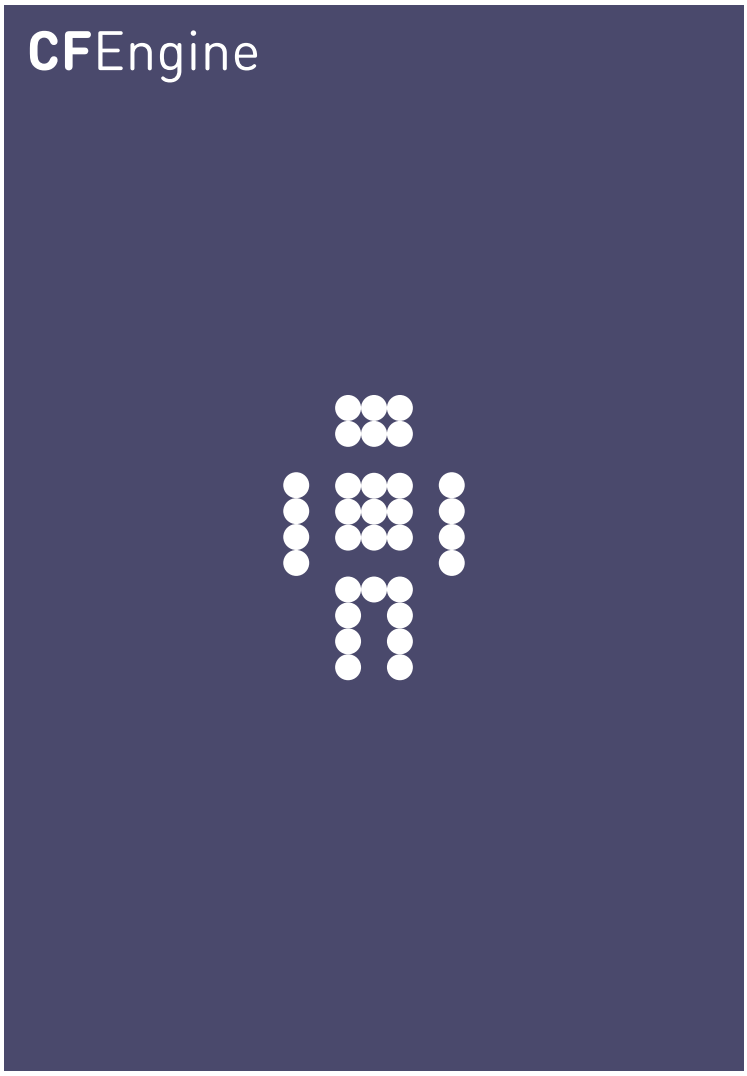

Content-Driven Policies

A CFEngine Special Topics Handbook

CFEngine AS

This document describes simplified policy writing using Content-Driven Policies in CFEngine Nova.

Table of Contents

What is a Content-Driven Policy?

A Content-Driven Policy is a text file with lines containing semi-colon separated fields, like a spreadsheet or tabular file. Each line in the file is parsed and results in a specific type of promise being made, depending on which type the Content-Driven Policy is. The '`services`' Content-Driven Policy is shown below.

```
# masterfiles/cdp_inputs/service_list.txt

Dnscache;stop;fix;windows
ALG;start;warn;windows
RemoteRegistry;start;fix;Windows_Server_2008
```

The meaning of the fields are different depending of the policy type, but explained in the file header. With these three lines, we ensure the correct status of three services on all our Windows machines and are given specialized reports on the outcome. The Content-Driven Policy services report is shown below.

Services (cdp_inputs/service_list.txt)

Host	Service name	Runstatus	Action	Class expression	State	Time checked
win-3squj7oeaku	RemoteRegistry	start	fix	Windows_Server_2008	Compliant	Tue Oct 19 15:31:20 2010
win-3squj7oeaku	Dnscache	stop	fix	windows	Compliant	Tue Oct 19 15:31:20 2010
win-3squj7oeaku	ALG	start	warn	windows	Not Compliant	Tue Oct 19 15:31:20 2010

Why should I use Content-Driven Policies?

As seen in the example above, Content-Driven Policies are easy to write and maintain, especially for users not very familiar with the CFEngine language. They are designed to capture the essence of a specific, popular use of CFEngine, and make it easier. For example, the services Content-Driven Policy above has the following equivalent in the CFEngine language.

```
bundle agent service_example
{
services:

  "Dnscache"
    comment            => "Check services status of Dnscache",
    handle             => "srv_Dnscache_windows",
    service_policy     => "stop",
    service_method     => force_deps,
    action             => policy("fix"),
    ifvarclass         => "windows";

  "ALG"
    comment            => "Check services status of ALG",
    handle             => "srv_ALG_windows",
```

```
      service_policy     => "start",
      service_method     => force_deps,
      action             => policy("warn"),
      ifvarclass         => "windows";

    "RemoteRegistry"
      comment            => "Check services status of ALG",
      handle             => "srv_ALG_windows",
      service_policy     => "start",
      service_method     => force_deps,
      action             => policy("fix"),
      ifvarclass         => "Windows_Server_2008";

}
```

Writing this policy is clearly more time-consuming and error-prone. On the other hand, it allows for much more flexibility than Content-Driven Policies, when that is needed.

CFEngine provides Content-Driven Policies to cover mainstream management tasks like the following.

- File change/difference management
- Service management
- Database management
- Application / script management

How do Content-Driven Policies work in detail?

The text files in `masterfiles/cdp_inputs/` (e.g. '`registry_list.txt`') are parsed into CFEngine lists by corresponding `cdp_*` files in `masterfiles/` (e.g. '`cdp_registry.cf`'). It is the latter set of files that actually implement the policies in the text files.

The Knowledge Map contains reports specifically designed to match the Content-Driven Policies.

Can I make my own Content-Diven Policies?

It is possible to mimic the structure of the existing Content-Driven Policies to implement new ones, for new purposes.

However, CFEngine AS will be creating more of these best-practice policies. Thus, making a feature request at CFEngine Support may result in your proposal being developed and supported by professionals at CFEngine AS. Furthermore, Knowledge Map reports currently need to be developed induvidually by CFEngine AS.