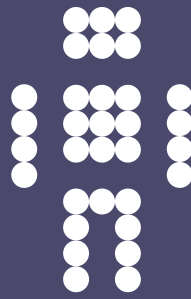


CFEngine



Supplement to CFEngine 3 Enterprise Evaluation Guide

CFEngine Enterprise Documentation

Updated 31. May 2012

CFEngine

This document supplements the 'CFEngine 3 Enterprise Evaluation Guide' with examples for all report types available in the CFEngine 3 Enterprise Mission Portal.

Copyright © 2012 CFEngine AS. The features described herein are in provided for user convenience and imply no warranty whatsoever to the extent of applicable law.

Table of Contents

1	Standard reports in CFEngine 3 Enterprise	1
1.1	Policy	2
1.1.1	Compliance by bundle	2
1.1.2	Compliance by promise	3
1.1.3	Promises not kept log	4
1.1.4	Promises not kept summary	5
1.1.5	Promises repaired log	6
1.1.6	Promises repaired summary	7
1.2	Accounting	8
1.2.1	Benchmarks	8
1.2.2	Business value	9
1.2.3	Compliance summary	10
1.3	System	11
1.3.1	Context classes	11
1.3.2	Last saw neighbours	12
1.3.3	Variables	14
1.3.4	Weakest Hosts	15
1.4	Software	15
1.4.1	Installed	15
1.4.2	Patches available	16
1.4.3	Patches installed	17
1.5	File watch	18
1.5.1	Change summary	18
1.5.2	Setuid	19
1.5.3	Text changes	20

1 Standard reports in CFEngine 3 Enterprise

A significant capability of CFEngine 3 Enterprise is automated system reporting: it collects history, state and change data about computers and ties them together. A report is a tabular summary of CFEngine's internal information, tailored to a particular purpose, searchable, and describes attributes and qualities of managed hosts.

Standard reports in CFEngine 3 Enterprise can be accessed through the **Reports tab** in the **Engineering room**. The drop down menus list all standard report categories and each category contains information about different aspects of the Mission. When you click one of them, the Mission Portal will present a query form that is adapted to the chosen report category. CFEngine uses regular expressions in these queries, for maximum flexibility and to minimize system impact. The details of these queries and the content of the resulting reports are outlined in the following sections.

Reports are updated at different intervals, the default values are every 5 minutes or every 6 hours. Below is a list of standard reports, updated every 5 minutes unless otherwise noted:

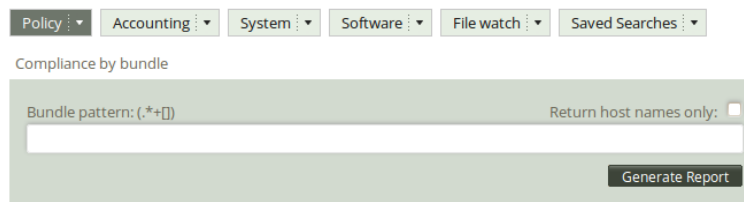
- Policy
 - Compliance by bundle: Status of promise bundles and when they were last verified
 - Compliance by promise: Compliance of each promise individually
 - Promises not kept log: Log of promises that could not or would not be kept
 - Promises not kept summary: Cumulative (histogram) summary of promises not kept
 - Promises repaired log: Log of actual repairs made to the system
 - Promises repaired summary: Cumulative (histogram) summary of promises repaired
- Accounting
 - Benchmarks: Execution time used to verify selected promises
 - Business value report: Accumulated value of promises kept (6 hrs)
 - Compliance summary: Total summary of host compliance
- System
 - Context classes: User defined classes observed on the system
 - Last saw neighbours: Log of when neighboring hosts were last observed online
 - Variables: Table of variable values last observed (6 hrs)
 - Weakest hosts: Ranking of hosts with poorest performing hosts on top
- Software
 - Installed: List of software packages claimed to be installed according to the local package manager
 - Patches available: A list of patches currently claimed to be available by the local package manager
 - Patch installed: A list of (un)applied patches according to the local package manager
- File watch
 - Change summary: Log of all detected changes to files from changes promises
 - Setuid: Current list of observed setuid/setgid root programs (6 hrs)
 - Text changes: Delta/difference comparison showing file changes

1.1 Policy

1.1.1 Compliance by bundle

Promises are the fundamental statements in CFEngine, they make up the definition of the desired state of a system. A *bundle* is a collection of promises in a 'sub-routine-like' container. The purpose of bundles is to allow greater flexibility to break down the contents of policies and give them names. Bundles also allow to re-use promise code by parameterizing it.

The 'Compliance by bundle' is useful for checking the compliance of specific bundles and when they were last verified. Put your mouse over the **Policy** drop down menu in the **Reports** tab, then select **Compliance by bundle**. The Compliance by bundle query can filter by bundle pattern (pattern in bundle name). Leaving the fields blank will result in a report listing all bundles in your policies.



Policy Accounting System Software File watch Saved Searches

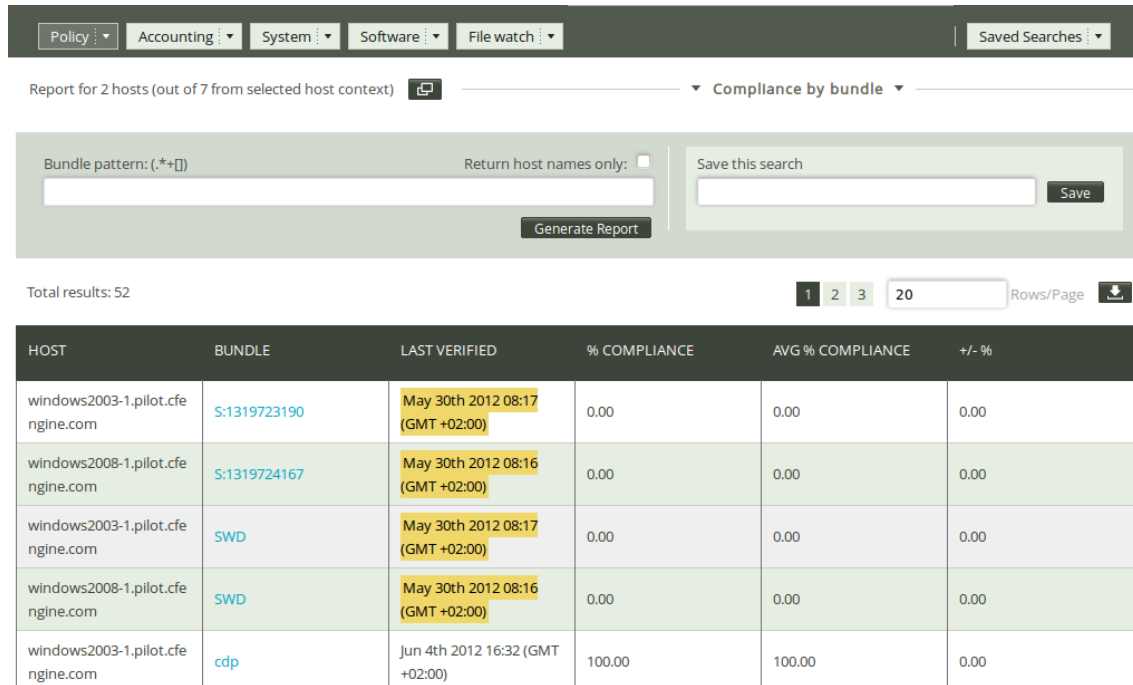
Compliance by bundle

Bundle pattern: (.*+[]) Return host names only:

Generate Report

Figure: Compliance by bundle query

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of bundles that suit the query criteria entered above. It displays the host names on which these bundles can be found ('Host'), the name of the bundles ('Bundle'), the time stamp at the moment of verification ('Last verified'), the time passed since last verification ('Last verified'), the current bundle compliance ('% compliance'), the average bundle compliance ('Avg % compliance') and the deviation of the current compliance from the average ('+/- %').



HOST	BUNDLE	LAST VERIFIED	% COMPLIANCE	AVG % COMPLIANCE	+/- %
windows2003-1.pilot.cfengine.com	S:1319723190	May 30th 2012 08:17 (GMT +02:00)	0.00	0.00	0.00
windows2008-1.pilot.cfengine.com	S:1319724167	May 30th 2012 08:16 (GMT +02:00)	0.00	0.00	0.00
windows2003-1.pilot.cfengine.com	SWD	May 30th 2012 08:17 (GMT +02:00)	0.00	0.00	0.00
windows2008-1.pilot.cfengine.com	SWD	May 30th 2012 08:16 (GMT +02:00)	0.00	0.00	0.00
windows2003-1.pilot.cfengine.com	cdp	Jun 4th 2012 16:32 (GMT +02:00)	100.00	100.00	0.00

Figure: Compliance by bundle

1.1.2 Compliance by promise

Promises are the fundamental statements in CFEngine, the policy atoms. Promises can be made about all kinds of different subjects, from file attributes, to the execution of commands, access control decisions and knowledge relationships. If there is no promise, nothing happens. It is considered compliant if CFEngine is able to keep the promise, and conversely, not compliant, or not kept, in the opposite case.

The 'Compliance by promise' report is useful for checking the current status of your system in high detail. You can find out which parts of a bundle work and which do not. The report also predicts the probability of compliance based on the history of specific promises, allowing you to assess the degree to which the problem is of a more transient or permanent nature.

Put your mouse over the **Policy** drop down menu in the **Reports** tab, then select **Compliance by promise**. The compliance by promise query can filter by (patterns in) promise handle and the compliance status of promises (any/compliant/repaired/non-compliant, drop-down menu).

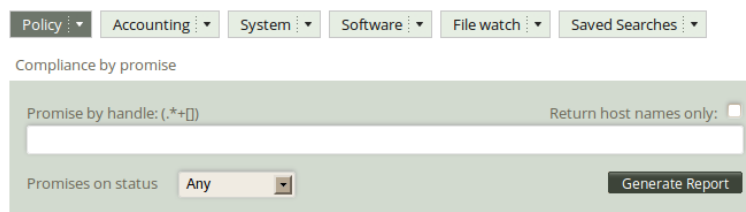


Figure: Compliance by promise query

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of hosts that suit the query criteria entered above. The result presents the host names ('Host'), the promise identifier 'Promise handle', 'Last known state' (compliant or not compliant), likelihood of a promise being

compliant ('% runs kept'), uncertainty of the likelihood ('+/- %', measured in one standard deviation of '% runs kept'), and the time stamp of the last time the promise was run ('Last verified').

HOST	PROMISE HANDLE	LAST KNOWN STATE	% RUNS KEPT	+/- %	LAST VERIFIED
ubuntu10-1.pilot.cfengine.com	cdp_cmd_bin_sleep_5_linux	Repaired	50.00	0.00	Jun 4th 2012 17:00 (GMT +02:00)
ubuntu10-1.pilot.cfengine.com	promise_cdp_cf_16	Compliant	100.00	0.00	Jun 4th 2012 17:00 (GMT +02:00)
ubuntu10-1.pilot.cfengine.com	cdp_chg_etc_shadow_linux_solarisx86	Compliant	100.00	0.00	Jun 4th 2012 16:59 (GMT +02:00)
ubuntu10-1.pilot.cfengine.com	cdp_diff_etc_group_linux_solarisx86	Compliant	100.00	0.00	Jun 4th 2012 16:59 (GMT +02:00)

Figure: Compliance by promise report

1.1.3 Promises not kept log

The Business room in the Mission Portal gives an overview of the general status of your system, including six hour summaries of promises kept, repaired, and not kept from the last week. The Promises not kept log is useful to get a complete overview of the history of promises not kept, including execution order and events that are more than a week old. Put your mouse over the **Policy** drop down menu in the **Reports** tab, then select **Promises not kept log**.

Figure: Promises not kept log query

The Promises not kept log query can filter by (patterns in) promise handles, cause (i.e. what was not kept), and a desired time interval. Leaving the fields blank will result in a report listing all promises that were not kept and the time of occurrence.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of promises that suit the query criteria entered above. The results are presented as 'Host' (host name), 'Promise

handle' (identifier of the promise), 'Report' (what was not kept), and 'Time' (time stamp of the event).

HOST	PROMISE HANDLE	REPORT	TIME
windows2003-1.pilot.cfengine.com	cdp_cmd_c_windows_system32_cmd_exe_c_eho_hello_windows_Hr11	!! Finished command related to promiser "c:\windows\system32\cmd.exe /c "eho hello"" -- an error occurred (returned 1)	Jun 4th 2012 11:58 (GMT +02:00)
windows2008-1.pilot.cfengine.com	cdp_cmd_c_windows_system32_cmd_exe_c_eho_hello_windows_Hr11	!! Finished command related to promiser "c:\windows\system32\cmd.exe /c "eho hello"" -- an error occurred (returned 1)	Jun 4th 2012 11:58 (GMT +02:00)
windows2003-1.pilot.cfengine.com	cdp_cmd_c_windows_system32_cmd_exe_c_eho_hello_windows_Hr11	!! Finished command related to promiser "c:\windows\system32\cmd.exe /c "eho hello"" -- an error occurred (returned 1)	Jun 4th 2012 11:57 (GMT +02:00)

Figure: Promises not kept log report

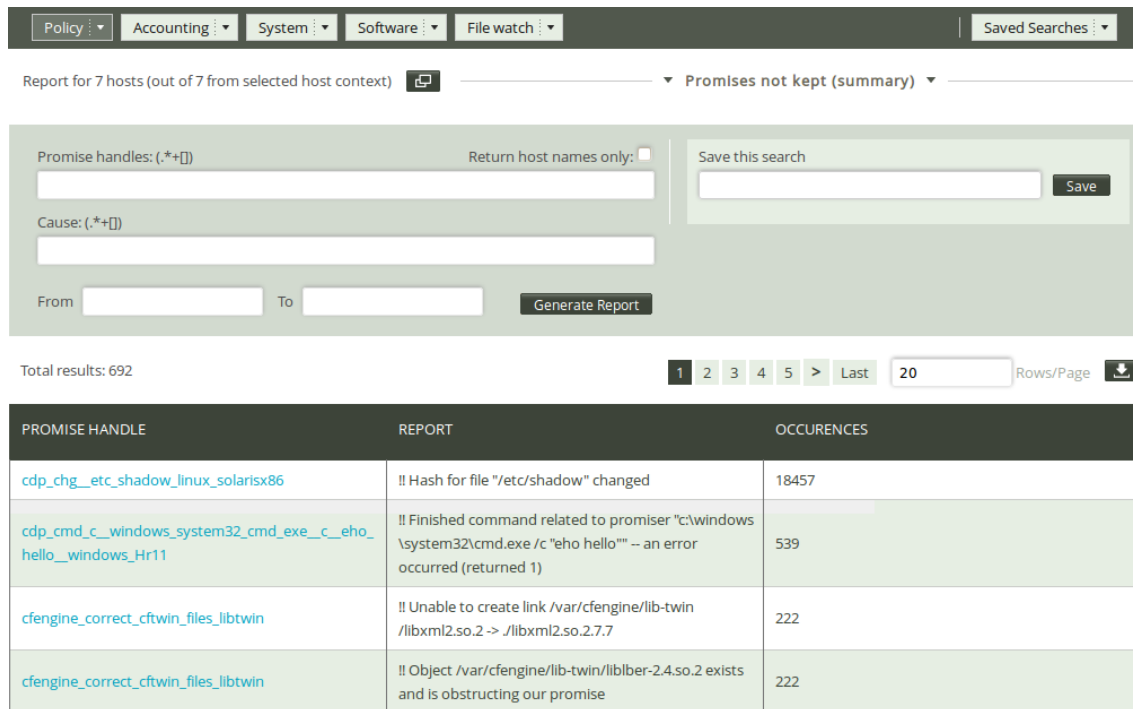
1.1.4 Promises not kept summary

If the Promises not kept log is too detailed for your needs, the Promises not kept summary report eliminates the time stamp of the promises repaired and presents a cumulative summary of promises repaired, i.e. the total number times a promise was not kept. Put your mouse over the **Policy** drop down menu in the **Reports** tab, then select **Promises not kept summary**.

Figure: Promises not kept summary query

The Promise not kept summary query can filter by (patterns in) promise handles, cause (i.e. what was not kept), and a desired time interval. Leaving the fields blank will result in a report listing all promises that were not kept and their cumulative number of occurrences.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of promises that suit the query criteria entered above. The results are presented as 'Promise handle', 'Report' (what was not kept), and 'Occurrences' (the number of times the promise was not kept).



PROMISE HANDLE	REPORT	OCCURENCES
cdp_chg_etc_shadow_linux_solarisx86	!! Hash for file "/etc/shadow" changed	18457
cdp_cmd_c_windows_system32_cmd_exe_c_eho_hello_windows_Hr11	!! Finished command related to promiser "c:\windows\system32\cmd.exe /c "eho hello" -- an error occurred (returned 1)	539
cfengine_correct_cftwin_files_libtwin	!! Unable to create link /var/cfengine/lib-twin/libxml2.so.2 -> /libxml2.so.2.7.7	222
cfengine_correct_cftwin_files_libtwin	!! Object /var/cfengine/lib-twin/liblber-2.4.so.2 exists and is obstructing our promise	222

Figure: Promises not kept summary report

1.1.5 Promises repaired log

The Business room in the Mission Portal gives an overview of the general status of your system, including six hour summaries of promises kept, repaired, and not kept from the last week. The Promises repaired log is useful to get a complete overview of the history of promises repaired, including execution order and events that are more than a week old. Put your mouse over the **Policy** drop down menu in the **Reports** tab, then select **Promises repaired log**.

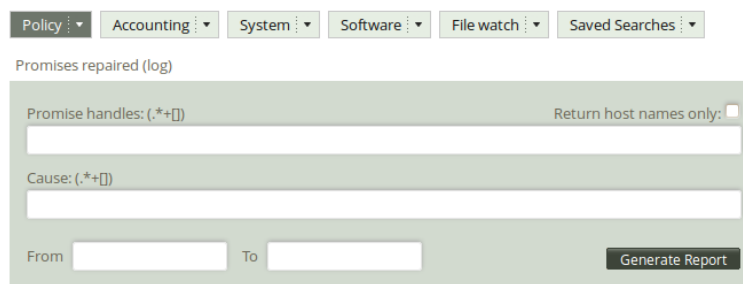
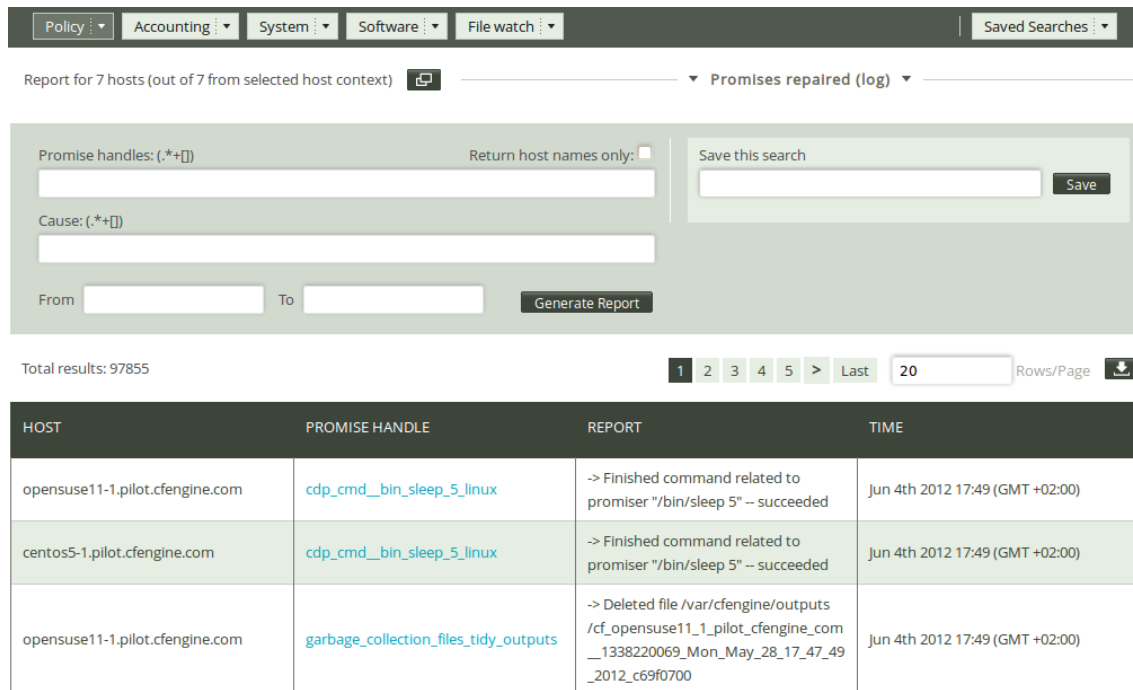


Figure: Promises repaired log query

The Promise repaired log query can filter by (patterns in) promise handles, cause (i.e. what was not kept), and a desired time interval. Leaving the fields blank will result in a report listing all promises that were repaired and the time of occurrence.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of promises that suit the query criteria entered above. The results are presented as 'Host' (host name), 'Promise handle' (identifier of the promise), 'Report' (what was repaired), and 'Time' (time stamp of the repair action).



Report for 7 hosts (out of 7 from selected host context) Promises repaired (log)

Promise handles: (.*)+[] Return host names only:

Cause: (.*)+[]

From To

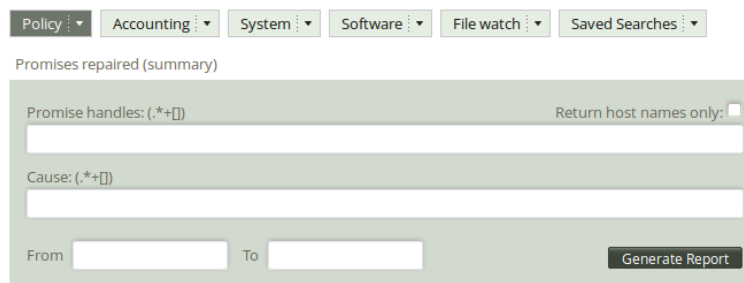
Total results: 97855 Rows/Page

HOST	PROMISE HANDLE	REPORT	TIME
opensuse11-1.pilot.cfengine.com	cdp_cmd_bin_sleep_5_linux	-> Finished command related to promiser "/bin/sleep 5" -- succeeded	Jun 4th 2012 17:49 (GMT +02:00)
centos5-1.pilot.cfengine.com	cdp_cmd_bin_sleep_5_linux	-> Finished command related to promiser "/bin/sleep 5" -- succeeded	Jun 4th 2012 17:49 (GMT +02:00)
opensuse11-1.pilot.cfengine.com	garbage_collection_files_tidy_outputs	-> Deleted file /var/cfengine/outputs /cf_opensuse11_1_pilot_cfengine_com__1338220069_Mon_May_28_17_47_49_2012_c69f0700	Jun 4th 2012 17:49 (GMT +02:00)

Figure: Promises repaired log report

1.1.6 Promises repaired summary

If the Promises repaired log is too detailed for your needs, the Promises repaired summary report eliminates the time stamp of the promises repaired and presents a cumulative summary of promises repaired, i.e. the total number times a promise has been repaired. Put your mouse over the **Policy** drop down menu in the **Reports** tab, then select **Promises repaired summary**.



Policy

Promises repaired (summary)

Promise handles: (.*)+[] Return host names only:

Cause: (.*)+[]

From To

Figure: Promises repaired summary query

The Promise repaired summary query can filter by (patterns in) promise handles, cause (i.e. what was repaired), and a desired time interval. Leaving the fields blank will result in a report listing all promises that were repaired and their cumulative number of occurrences.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of promises that suit the query criteria entered above. The results are presented as 'Promise handle' (identifier of the promise), 'Report' (what was repaired), and 'Occurrences' (number of occurrences of repair).

Report for 7 hosts (out of 7 from selected host context) **Promises repaired (summary)**

Promise handles: (.*+[]) Return host names only:

Cause: (.*+[])

From To **Generate Report**

Total results: 40883 1 2 3 4 5 > Last Rows/Page

PROMISE HANDLE	REPORT	OCCURENCES
cdp_cmd_bin_sleep_5_linux	-> Finished command related to promiser "/bin/sleep 5" -- succeeded	23544
cdp_cmd_c_windows_system32_cmd_exe_c_echo_hello_succeeded_c_reportfile_txt_hello_failed_windows	-> Finished command related to promiser "c:\windows\system32\cmd.exe /c "echo hello succeeded > c:\reportfile.txt"" -- succeeded	11978
cdp_cmd_c_windows_system32_cmd_exe_c_echo_hello_windows_Hr11	-> Finished command related to promiser "c:\windows\system32\cmd.exe /c "echo hello"" -- succeeded	11439

Figure: Promises repaired summary report

1.2 Accounting

1.2.1 Benchmarks

CFEngine 3 Enterprise uses several monitoring probes to reflect on general system performance¹. One probe looks at the time it takes to execute selected promises; results are summarized in the 'Benchmark' report. The user can thus evaluate which parts of a policy put charge on the system in terms of time spent completing a task. Longer tasks, such as command execution and file copying, are measured by default, but other tasks have to be measured explicitly by stating so in a policy. Note however that most promises made in CFEngine are executed so fast we are not able to measure the time it takes to complete them.

Put your mouse over the **Accounting** drop down menu in the **Reports** tab, then select **Benchmarks**.

¹ See also section on Vital signs in the CFEngine 3 Enterprise Owner's Manual.

Figure: Benchmarks query

The Benchmark query can filter by (patterns in) job name. Leaving the field blank will result in a report listing the performance of all monitored jobs.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of events that suit the query criteria entered above. 'Host' (host name), 'Event' (job name), 'Last completion time' (most recent performance value, i.e. the time it took to complete the job), 'Avg completion time' (average of all Last completion time), '+/- seconds' (difference expressed as one standard deviation of 'Avg completion time'), and 'Last performed' (time stamp of last execution).

HOST	EVENT	LAST COMPLETION TIME (SECONDS)	AVG COMPLETION TIME (SECONDS)	+/- SECONDS	LAST PERFORMED
solaris10-1.pilot.cfengine.com	Copy(localhost:/var/cfengine/bin > /usr/local/sbin)	0.00	0.00	0.00	Jun 4th 2012 18:25 (GMT +02:00)
solaris10-1.pilot.cfengine.com	Copy(172.20.10.1:/var/cfengine/masterfiles/cf_promises_validated > /var/cfengine/inputs/cf_promises_validated)	0.00	0.00	0.00	Jun 4th 2012 18:25 (GMT +02:00)
solaris10-1.pilot.cfengine.com	Copy(172.20.10.1:/var/cfengine/master_software_updates/sunos_5.10_i86pc > /var/cfengine/software_updates/sunos_5.10_i86pc)	0.01	0.01	0.00	Jun 4th 2012 18:25 (GMT +02:00)

Figure: Benchmarks report

1.2.2 Business value

One of the capabilities of CFEngine 3 Enterprise is to add business or organizational value to the configuration model. The notion of business value is not necessarily a clear concept, but a simple approach is to attach a monetary value to the outcome of promises.

CFEngine's default values for promises kept, promises repaired and promises not kept are 1, 0.5, and -1, respectively. The values are summed and recorded at a set time interval, and the results are summarized for every host and day.

Put your mouse over the **Accounting** drop down menu in the **Reports** tab, then select **Business value**.

Policy ▾ Accounting ▾ System ▾ Software ▾ File watch ▾ Saved Searches ▾

Business value

Date: Return host names only:

Figure: Business value query

The Business value query can filter by date. Leaving the field blank will result in a report listing the business value of all promises that have had value attached to them over selected hosts and days.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of hosts that suit the query criteria entered above. The result presents each host name ('Host'), date ('Summary of Day'), and the sum of the value of the promises kept ('Value of promises kept'), repaired ('Value of repairs'), and not kept ('Loss for promises not kept').

Policy ▾ Accounting ▾ System ▾ Software ▾ File watch ▾ Saved Searches ▾

Report for 7 hosts (out of 7 from selected host context) Business value ▾

Date: Return host names only: Save this search

Total results: 50 1 2 3 20 Rows/Page

HOST	SUMMARY OF DAY	VALUE OF PROMISES KEPT	VALUE OF REPAIRS	LOSS FOR PROMISES NOT KEPT
centos5-1.pilot.cfengine.com	5 Jun 2012	561845	426.5	-2
policyhub-1.pilot.cfengine.com	5 Jun 2012	2831471	635	0
windows2008-1.pilot.cfengine.com	5 Jun 2012	11478	644.5	-90

Figure: Business value report

1.2.3 Compliance summary

CFEngine policies are collections of promises contained in a text file, they are the CFEngine scripts that define what state you want your system to be in. The compliance summary report gives an overview of policy status. It shows the current status of your system in a coarse manner, allowing you to quickly identify which areas need follow-up. Put your mouse over the **Accounting** drop down menu in the **Reports** tab, then select **Compliance summary**.

Figure: Compliance summary query

The compliance summary query can filter by version of your policy files (version number is maintained manually in policy). Leaving the field blank will result in a report listing all hosts and policies.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of hosts that suit the query criteria entered above. The result presents the host names ('Host'), policy version ('Policy version'), percentage of promises kept within the listed policies ('% kept'), percentage of promises repaired within the listed policies ('% repaired'), percentage of promises not kept within the listed policies ('% not kept'), and the time stamp of the last status check ('Last verified').

HOST	POLICY VERSION	% KEPT	% REPAIRED	% NOT KEPT	LAST VERIFIED
opensuse11-1.pilot.cfengine.com	Failsafe.cf 2.1.0 (agent-0) Promises.cf 2.1.0 (agent-0)	100	0	0	Jun 5th 2012 15:27 (GMT +02:00)
policyhub-1.pilot.cfengine.com	Failsafe.cf 2.1.0 (agent-0) Promises.cf 2.1.0 (agent-0)	99	1	0	Jun 5th 2012 15:26 (GMT +02:00)

Figure: Compliance summary report

1.3 System

1.3.1 Context classes

CFEngine classes are certain true/false (Boolean) propositions that determine in what context, or setting, a promise is made. Each time CFEngine runs (by default every five minutes), it discovers and classifies properties of the environment in which it runs. These discovered properties are called 'hard classes' and cannot be changed by users. CFEngine also operates with soft classes, i.e. user-defined types.

The Context classes report is useful for looking at hosts in specific contexts, for instance to find out which machines run on windows. Put your mouse over the **System** drop down menu in the **Reports** tab, then select **Context classes**.

Figure: Context classes query

The context classes query can filter by (pattern in) class name. Leaving the field blank will result in a report listing all hosts and classes.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of hosts that suit the query criteria entered above. The result presents the host names ('Host'), 'Class or context', probability of occurrence ('In % runs'), uncertainty on the probability '+/- %' (standard deviation of 'In % runs'), and the last time the class was observed ('Last occurred').

Report for 7 hosts (out of 7 from selected host context) Context classes

Class pattern: (.*) Return host names only: Save this search Save

Generate Report

Total results: 1204 1 2 3 4 5 > Last 20 Rows/Page

HOST	CLASS OR CONTEXT	IN % RUNS	+/- %	LAST OCCURRED
policyhub-1.pilot.cfengine.com	10_10_150_52	100	0	Jun 1st 2012 15:58 (GMT +02:00)
policyhub-1.pilot.cfengine.com	10_10_150_58	100	0	Jun 5th 2012 15:40 (GMT +02:00)
policyhub-1.pilot.cfengine.com	172_20_10_1	100	0	Jun 5th 2012 15:40 (GMT +02:00)

Figure: Context classes report

1.3.2 Last saw neighbours

Sometimes it is not possible to connect to a machine under CFEngine's management, either due to network errors or temporary lack of network entirely (for instance on ships at sea or submarines). CFEngine 3 Enterprise's Mission Portal monitors all connections, incoming and outgoing, between all managed hosts, and creates a log of when neighboring hosts were last observed online. This information is used to set the host availability status and, through analysis of the connection history, give an idea of the regularity of connections between hosts.

The Last saw neighbours report is useful for checking the communication pattern between managed hosts and when they last were in touch with each other. Put your mouse over the **System** drop down menu in the **Reports** tab, then select **Last saw neighbours**.

Figure: Last saw neighbours query

The Last saw neighbours query can filter by (patterns in) remote host name, remote host IP address, remote host key, and minimum hours ago (since the last connection was made). Leaving the fields blank will result in a report listing all connections made to and from the managed machines (including the hub).

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of all communication that suits the query criteria entered above. Every connection is logged on the concerned nodes as incoming (Initiated by them) or outgoing (Initiated by us), the same connection will therefore appear twice in the report (once for each node). The results are presented in the following column format: 'Host' (host name), 'Comms initiated' (identifies whether the connection is incoming (by them (-)) or outgoing (by us (+))), 'Remote host name', 'Remote IP address', 'Was last seen at' (time stamp of the connection), 'Hrs ago' (interval between current time and Last seen), 'Avg comms interval' (average time between each connection), Uncertainty '+/- hrs' (standard deviation of Average interval), and 'Remote host's key' (identifying key of the remote host).

HOST	COMMS INITIATED	REMOTE HOST NAME	REMOTE IP ADDRESS	WAS LAST SEEN AT	HRS AGO	AVG COMMS INTERVAL	+/- HRS	REMOTE HOST'S KEY
policyhub-1.pilot.cfengine.com	by us (+)	policyhub-1.pilot.cfengine.com	172.20.10.1	Jun 5th 2012 15:55 (GMT +02:00)	0.00	0.08	0.00	SHA=2f8d59051aac6a2a9538028b2cc8768e69c4db8afc4be50fc6ccbd92336b4e6
policyhub-1.pilot.cfengine.com	by them (-)	policyhub-1.pilot.cfengine.com	172.20.10.1	Jun 5th 2012 15:55 (GMT +02:00)	0.00	0.08	0.00	SHA=2f8d59051aac6a2a9538028b2cc8768e69c4db8afc4be50fc6ccbd92336b4e6

Figure: Last saw neighbours report

1.3.3 Variables

The 'Variables report' is useful for tracking your variables and checking their values, for instance to see if they behave in the expected manner. Put your mouse over the **System** drop down menu in the **Reports** tab, then select **Variables**.

Figure: Variables query

The Variables query can filter by (patterns in) scope (bundle where the variable is used), Lvalue (name of variable), Rvalue (content of variable), and type (string, integer, etc.). Leaving the fields blank will result in a report listing all variables that were last observed on the system.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of variables that suit the query criteria entered above. The results are presented in table form/blocks of scope (i.e. in which bundle the variables appear) with the following column format: 'Host' (name of host where the variable is defined), 'Type' (type of the variable), 'Name', 'Value' and 'Last seen'.

Report for 7 hosts (out of 7 from selected host context) Variables

Total results: 2897 1 2 3 4 5 > Last Rows/Page

bundle update_svn: 1 variables

HOST	TYPE	NAME	VALUE	LAST SEEN
solaris10-1.pilot.cfengine.com	string	master_location	/var/cfengine/masterfiles	Jun 5th 2012 13:59 (GMT +02:00)

bundle update_bins: 7 variables

HOST	TYPE	NAME	VALUE	LAST SEEN
solaris10-1.pilot.cfengine.com	string	stop_signal	term	Jun 5th 2012 13:59 (GMT +02:00)
solaris10-1.pilot.cfengine.com	string	pkgarch	i386	Jun 5th 2012 13:59 (GMT +02:00)

Figure: Variables report

1.3.4 Weakest Hosts

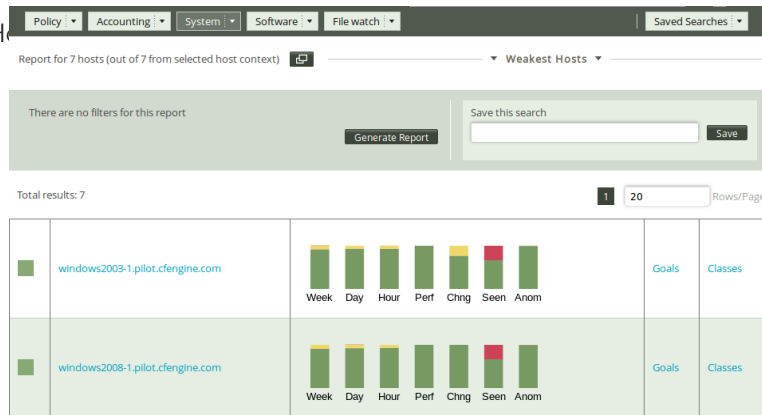


Figure: Weakest hosts report

1.4 Software

1.4.1 Installed

The 'Installed report' will list the software packages claimed to be installed according to the local package manager. Put your mouse over the **Software** drop down menu in the **Reports** tab, then select **Installed**.

The screenshot shows the 'Software installed query' form in the CFEngine 3 Enterprise interface. At the top, there are navigation tabs for Policy, Accounting, System, Software, File watch, and Saved Searches. Below the tabs, the form title is 'Installed'. The form has three input fields: 'Name:', 'Version:', and 'Architecture:'. There is a checkbox labeled 'Return host names only:'. A 'Generate Report' button is located at the bottom right of the form.

Figure: Software installed query

The Software installed query can filter by (patterns in) software name, version, and architecture. Leaving the fields blank will result in a report listing all hosts and software installed on the system.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview that suits the query criteria entered above. The results are presented as 'Host' (host name), 'Package name' (of software package), 'Version' (of software package), 'Architecture' (of machine on which software runs) and 'Last seen'.

Report for 6 hosts (out of 7 from selected host context) Installed

Name: Return host names only:

Version:

Architecture:

Save this search Save

Generate Report

Total results: 2831 1 2 3 4 5 > Last 20 Rows/Page

HOST	PACKAGE NAME	VERSION	ARCHITECTURE	LAST SEEN
solaris10-1.pilot.cfengine.com	BRCMbnx	11.10.0_REV=2007.06.20.13.12	i386	Jun 5th 2012 10:29 (GMT +02:00)
solaris10-1.pilot.cfengine.com	CADP160	1.21_REV=2005.01.17.23.31	i386	Jun 5th 2012 10:29 (GMT +02:00)

Figure: Software installed report

1.4.2 Patches available

Software packaging is a core paradigm in operating system release management today, and CFEngine supports a generic approach to integration with native operating support for packaging. Package promises allow CFEngine to make promises the state of software packages conditionally, given the assumption that a native package manager will perform the actual manipulations. Since no agent can make unconditional promises about another, this is the best that can be achieved.

Some package systems also support the idea of patches. These might be formally different objects to packages; a patch might contain material for several packages and be numbered differently. When you select patching-policy, the package name can be a regular expression that will match possible patch names, otherwise identifying specific patches can be cumbersome.

The patches available report is useful to get an overview of patches claimed to be available by the local package manager. Put your mouse over the **Software** drop down menu in the **Reports** tab, then select **Patches available**.

Policy Accounting System **Software** File watch Saved Searches

Patches available

Package name: (. *+[]) Return host names only:

Package version: (. *+[])

Package architecture:

Generate Report

Figure: Patches available query

The Patches available query can filter by (patterns in) package name, package version, and package architecture. Leaving the fields blank will result in a report listing all patches that can be installed on the system.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of patches that suit the query criteria entered above. The report presents the following columns: 'Host' (host name), 'Package name', 'Version' (patch version), and 'Architecture'.

HOST	PACKAGE NAME	VERSION	ARCHITECTURE
opensuse11-1.pilot.cfengine.com	Name	Version	default
opensuse11-1.pilot.cfengine.com	SuSEfirewall2	4330	default
opensuse11-1.pilot.cfengine.com	aaa_base	4171	default

Figure: Patches available report

1.4.3 Patches installed

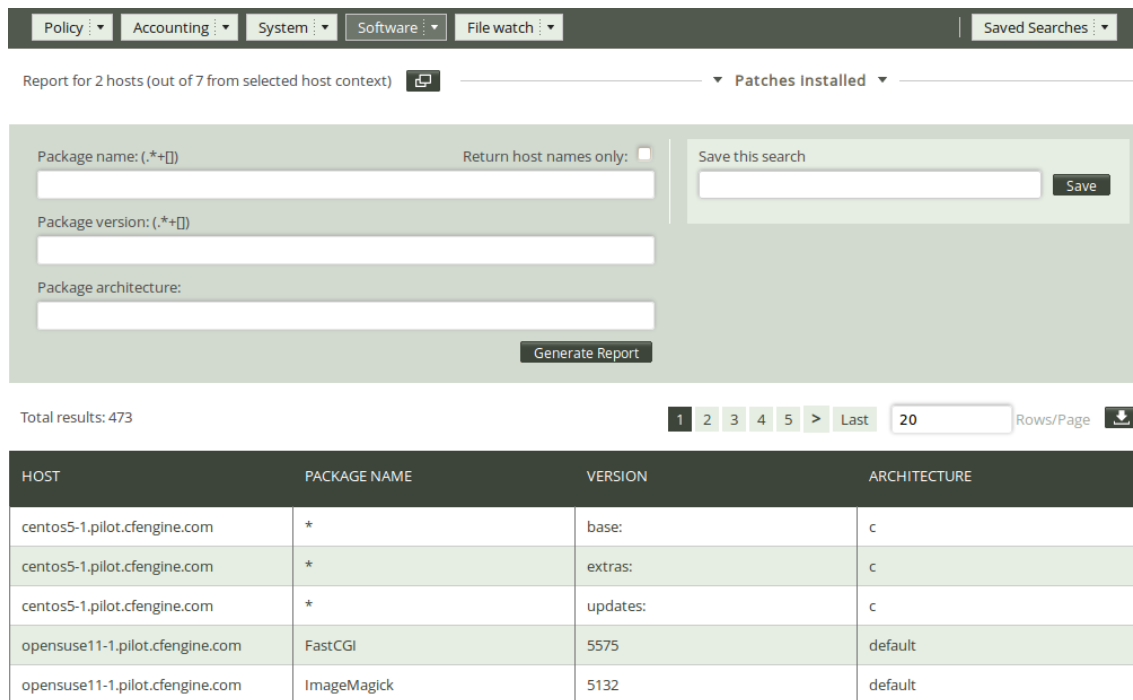
Patch management can be a delicate business: sometimes a patch can cause new problems, or perhaps even more problems than it fixes. IT managers therefore often like to be in control of what patches are applied to a system. The Patch status report gives system administrators a complete overview of applied patches according to the local package manager, and, in conjunction with the patches available report, allows them to consciously decide which patches to apply or not.

Put your mouse over the **Software** drop down menu in the **Reports** tab, then select **Patches installed**.

Figure: Patches installed query

The Patches installed query can filter by (patterns in) package name, package version, and package architecture. Leaving the fields blank will result in a report listing all patches applied to the system.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of patches that suit the query criteria entered above. The report presents results in the same format as the Patches available report: 'Host' (host name), 'Package name', 'Version' (patch version), and 'Architecture'.



The screenshot shows the CFEngine 3 Enterprise interface for the 'Patches installed' query. At the top, there are navigation tabs: Policy, Accounting, System, Software, File watch, and Saved Searches. Below these, a breadcrumb trail reads 'Report for 2 hosts (out of 7 from selected host context) > Patches Installed'. The main form contains three input fields: 'Package name: (.*)', 'Package version: (.*)', and 'Package architecture:'. There is a 'Return host names only:' checkbox and a 'Save this search' section with a 'Save' button. A 'Generate Report' button is at the bottom of the form. Below the form, it shows 'Total results: 473' and a pagination control with buttons for 1, 2, 3, 4, 5, and Last, along with a '20 Rows/Page' dropdown and a download icon. The resulting report is a table with the following data:

HOST	PACKAGE NAME	VERSION	ARCHITECTURE
centos5-1.pilot.cfengine.com	*	base:	c
centos5-1.pilot.cfengine.com	*	extras:	c
centos5-1.pilot.cfengine.com	*	updates:	c
opensuse11-1.pilot.cfengine.com	FastCGI	5575	default
opensuse11-1.pilot.cfengine.com	ImageMagick	5132	default

Figure: Patch status report

1.5 File watch

1.5.1 Change summary

File change monitoring is about detecting when file information on a computer system changes. Awareness of changes is often considered a major part of management, especially if they are unexpected or inadvertent (expected changes are usually not a problem). With CFEngine you can either set a general tripwire to detect all changes, or, in the case of the 'Change summary', monitor specific files with changes promises. The report gives you relative detail of change as it presents the name of files that have been changed, when they were changed and on what host they were changed.

Put your mouse over the **File watch** drop down menu in the **Reports** tab, then select **Change summary**.

Figure: File change summary query

The file change log query can filter by (patterns in) file name and time interval for when the change took place. Leaving the fields blank will result in a report listing changes detected on all monitored hosts and policies.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of hosts that suit the query criteria entered above. The result presents the host names ('Host'), name of the file where a change was detected ('File'), and time stamp of change detection ('Change detected at'). You can add your personal note in the right column, documenting any thoughts or issues that you might have about the query result.

File change log

PDF [Select host](#) [Select report](#) [Save this search](#) [New search](#)

Total results found: 963

HOST	FILE	CHANGE DETECTED AT	NOTE
ubuntu1.test.cfengine.com	!! File /var/cfengine/inputs /OrionCloud /Orion_Cloud_Pack_Demo.png was not in sha512 database - new file found	September 29, 2011 15:15:31 (GMT+2)	notes
ubuntu1.test.cfengine.com	!! File /var/cfengine/inputs /OrionCloud/.svn/entries was not in sha512 database - new file found	September 29, 2011 15:15:31 (GMT+2)	notes
ubuntu1.test.cfengine.com	!! File /var/cfengine/inputs /OrionCloud/.svn/text-base/c_cpp_env.cf.svn-base was not in sha512 database - new file found	September 29, 2011 15:15:31 (GMT+2)	notes

Figure: File change log report

1.5.2 Setuid

`setuid` and `setgid` (short for "set user ID upon execution" and "set group ID upon execution", respectively) are Unix access right flags that allow users to run an executable with the permissions of the executable's owner or group. They are often used to allow users on a computer system to run programs with temporarily elevated privileges in order to perform a specific task. The 'Setuid report' is useful to get an overview of what processes have been elevated to root privileges and potentially uncover security issues.

Put your mouse over the **File watch** drop down menu in the **Reports** tab, then select **Setuid**.

Figure: Setuidquery

The Setuid/gid root programs query can filter by (patterns in) file name. Leaving the fields blank will result in a report listing all hosts and files that have their permissions elevated to root.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of promises that suit the query criteria entered above. The results are presented as host name and files that have their permissions elevated to root.

HOST	FILE
solaris10-1.pilot.cfengine.com	/usr/bin/write
solaris10-1.pilot.cfengine.com	/usr/bin/volrmount
solaris10-1.pilot.cfengine.com	/usr/bin/tsoljdslabel
solaris10-1.pilot.cfengine.com	/usr/bin/su

Figure: Setuid report

1.5.3 Text changes

A diff is a file comparison utility that outputs the differences between two files. It is typically used to show the changes between one version of a file and a former version of the same file. Diff displays the changes made per line for text files. Once a file change has been identified, for instance as seen in the file change log, you can browse the details of that change in a file change diff report.

Put your mouse over the **File watch** drop down menu in the **Reports** tab, then select **Text changes**.

Figure: Text changes query

The Text changes query can filter by (pattern in) file name, (pattern in) content, and time interval for when the change took place.. Leaving the fields blank will result in a report listing changes detected on all monitored hosts and and policies.

Once you have clicked **Generate report**, CFEngine 3 Enterprise will list an overview of hosts that suit the query criteria entered above. The result presents the host names ('Host'), name of the file where a change was detected ('File'), the time stamp of change detection ('Change detected at'), and the actual diff (whether a line was added or deleted, the line number, and the content of the change; 'Change added (+), deleted (-); Line no; Content ').

File change diffs

  [Select host](#) [Select report](#)

[Save this search](#)
[New search](#)

Total results found : 6

HOST	FILE	CHANGE DETECTED AT	CHANGE ADDED(+), DELETED(-), LINE NO, CONTENT												
hub.test.cfengine.com	/tmp/test	September 26, 2011 14:35:28 (GMT+2)	<table border="1"> <tr> <td>-</td> <td>1</td> <td>hello</td> </tr> <tr> <td>-</td> <td>3</td> <td>bishwa</td> </tr> <tr> <td>-</td> <td>5</td> <td>next line</td> </tr> <tr> <td>+</td> <td>1</td> <td>test</td> </tr> </table>	-	1	hello	-	3	bishwa	-	5	next line	+	1	test
-	1	hello													
-	3	bishwa													
-	5	next line													
+	1	test													
policy.test.cfengine.com	/etc/group	September 26, 2011 13:20:59 (GMT+2)	<table border="1"> <tr> <td>+</td> <td>52</td> <td>ssl-cert: x:112:</td> </tr> </table>	+	52	ssl-cert: x:112:									
+	52	ssl-cert: x:112:													
hub.test.cfengine.com	/tmp/test	September 26, 2011 10:11:04 (GMT+2)	<table border="1"> <tr> <td>+</td> <td>5</td> <td>next line</td> </tr> </table>	+	5	next line									
+	5	next line													

Figure: File change diffs report