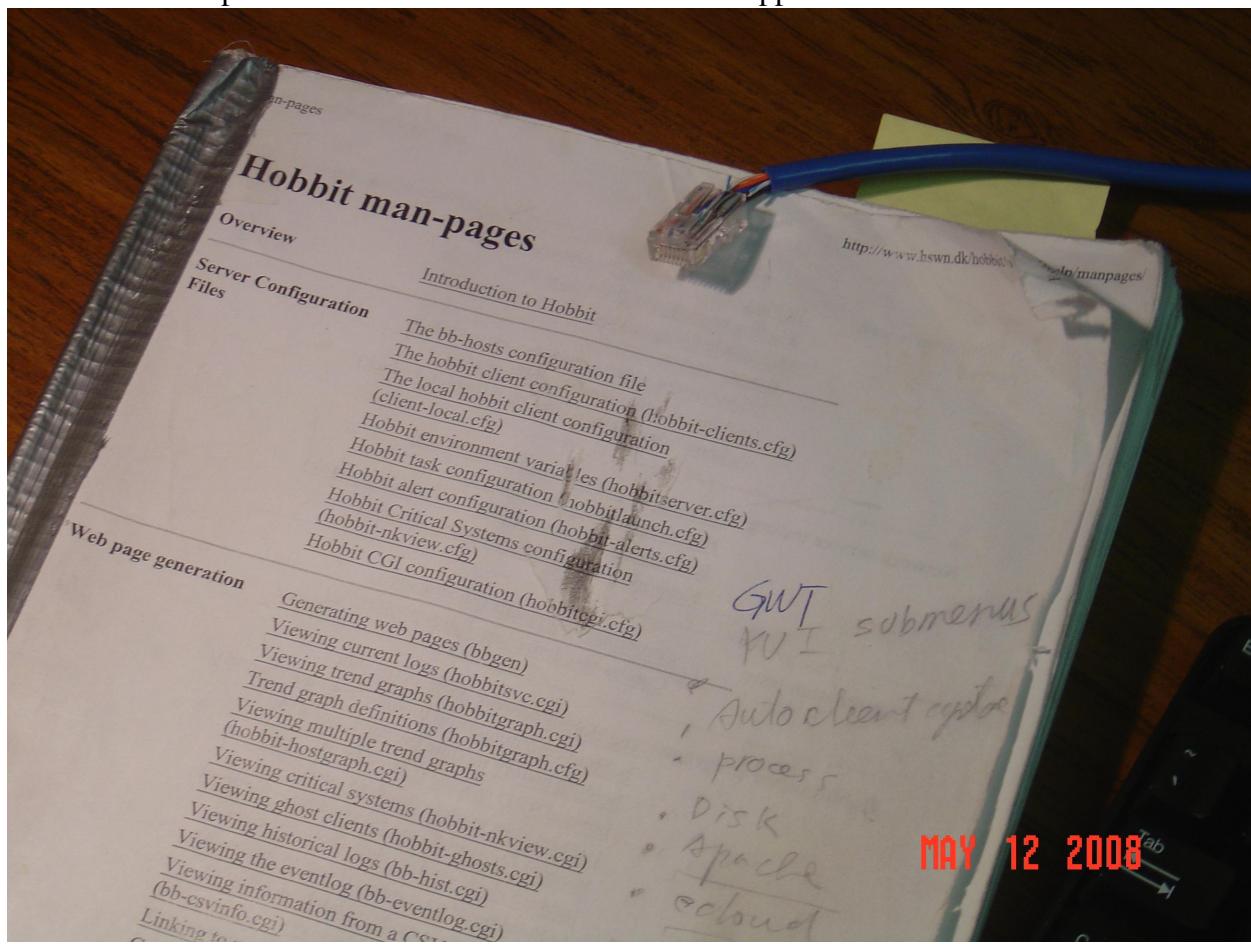


Hobbitmon Reference Manual

Hobbitmon Community

Copyright © 2008 GNU

This material may be distributed only subject to the terms and conditions set forth in version 1.0 of the Open Publication License. Please refer to Appendix B for the license text.



Contents

Contents	i
1 Preface	2
1.1 This book is a work in progress	2
1.2 Hobbit Documentation Road Map	2
1.3 Revision History	2
1.4 Colophon—this book is Free	3
2 Hobbit Overview	4
2.1 Hobbit - Introduction to the Hobbit Monitor	4
2.2 FEATURES	4
2.3 SECURITY	6
2.4 DEMONSTRATION SITE	6
2.5 PREREQUISITES	6
2.6 INSTALLATION	6
2.7 SUPPORT and MAILING LISTS	7
2.8 Hobbit Wiki Book	7
2.9 HOBBIT SERVER DAEMONS	7
2.10 HOBBIT NETWORK TEST TOOLS	8
2.11 HOBBIT TOOLS HANDLING THE WEB USER-INTERFACE	8
2.12 CLIENT-SIDE TOOLS	9
2.13 HOBBIT COMMUNICATION TOOLS	9
2.14 OTHER TOOLS	9
2.15 VERSIONS	10
2.16 COPYRIGHT	10
2.17 SEE ALSO	10
3 Configuring Hobbit Monitoring	13
3.1 Configuring Hobbit Monitoring	13
3.2 Setting up monitoring of hosts	14
3.3 Upgrading from Big Brother to Hobbit	17
4 Integrating MRTG data into Hobbit	19
4.1 Integrating MRTG data into Hobbit	19
5 Known issues in Hobbit	21
5.1 Known issues in Hobbit	21

6 Report Generation	24
6.1 HOBBIT-CONFREPORT.CGI	25
6.1.1 SYNOPSIS	25
6.1.2 DESCRIPTION	25
6.1.3 OPTIONS	25
6.1.4 BUGS	25
6.1.5 SEE ALSO	25
6.2 BB-REP.CGI	26
6.2.1 SYNOPSIS	26
6.2.2 DESCRIPTION	26
6.2.3 OPTIONS	26
6.2.4 FILES	26
6.2.5 ENVIRONMENT VARIABLES	26
6.2.6 SEE ALSO	27
6.3 HOBBITD_FILESTORE	28
6.3.1 SYNOPSIS	28
6.3.2 DESCRIPTION	28
6.3.3 OPTIONS	28
6.3.4 FILES	29
6.3.5 SEE ALSO	29
7 Network Service Testing	30
7.1 BBTEST-NET	30
7.1.1 SYNOPSIS	30
7.1.2 DESCRIPTION	30
7.1.3 GENERAL OPTIONS	30
7.1.4 OPTIONS FOR TESTS OF THE SIMPLE TCP SERVICES	32
7.1.5 OPTIONS FOR THE PING TEST	32
7.1.6 OPTIONS FOR HTTP (WEB) TESTS	32
7.1.7 OPTIONS FOR SSL CERTIFICATE TESTS	32
7.1.8 DEBUGGING OPTIONS	33
7.1.9 INFORMATIONAL OPTIONS	33
7.1.10 USING COOKIES IN WEB TESTS	33
7.1.11 ABOUT SSL CERTIFICATE CHECKS	33
7.1.12 LDAP TESTS	33
7.1.13 USING MULTIPLE NETWORK TEST SYSTEMS	35
7.1.14 BBTEST-NET INTERNALS	35
7.1.15 ENVIRONMENT VARIABLES	35
7.1.16 FILES	36
7.1.17 SEE ALSO	36
7.2 BB-SERVICES	37
7.2.1 SYNOPSIS	37
7.2.2 DESCRIPTION	37
7.2.3 FILE FORMAT	37
7.2.4 FILES	37
7.2.5 SEE ALSO	38
7.3 HOBBITFETCH	39
7.3.1 NAME	39
7.3.2 SYNOPSIS	39
7.3.3 DESCRIPTION	39
7.3.4 OPTIONS	39
7.3.5 SEE ALSO	39

7.4	bbcmd	40
7.4.1	SYNOPSIS	40
7.4.2	DESCRIPTION	40
7.4.3	SEE ALSO	40
7.5	CLIENTLAUNCH.CFG	41
7.5.1	SYNOPSIS	41
7.5.2	DESCRIPTION	41
7.5.3	FILE FORMAT	41
7.5.4	SEE ALSO	41
8	Combination Tests	42
8.1	BBCOMBOTEST	42
8.1.1	SYNOPSIS	42
8.1.2	DESCRIPTION	42
8.1.3	OPTIONS	42
8.1.4	FILES	42
8.1.5	SEE ALSO	42
8.2	HOBBITGRAPH.CGI	43
8.2.1	NAME	43
8.2.2	SYNOPSIS	43
8.2.3	DESCRIPTION	43
8.2.4	OPTIONS	43
8.2.5	ENVIRONMENT	43
8.2.6	FILES	43
8.2.7	SEE ALSO	44
8.3	BBCOMBOTEST.CFG	45
8.3.1	NAME	45
8.3.2	SYNOPSIS	45
8.3.3	DESCRIPTION	45
8.3.4	FILE FORMAT	45
8.3.5	EXAMPLE	45
8.3.6	FILES	46
8.3.7	SEE ALSO	46
8.4	HOBBITGRAPH.CGI	47
8.4.1	NAME	47
8.4.2	SYNOPSIS	47
8.4.3	DESCRIPTION	47
8.4.4	OPTIONS	47
8.4.5	ENVIRONMENT	47
8.4.6	FILES	47
8.4.7	SEE ALSO	48
8.5	BB-SNAPSHOT.CGI	49
8.5.1	NAME	49
8.5.2	SYNOPSIS	49
8.5.3	DESCRIPTION	49
8.5.4	OPTIONS	49
8.5.5	ENVIRONMENT VARIABLES	49
8.5.6	SEE ALSO	49
8.6	HOBBITD_SAMPLE	50
8.6.1	NAME	50
8.6.2	SYNOPSIS	50
8.6.3	DESCRIPTION	50

8.6.4	OPTIONS	50
8.6.5	SEE ALSO	50
8.7	BB-CSVINFO.CGI	51
8.7.1	NAME	51
8.7.2	SYNOPSIS	51
8.7.3	DESCRIPTION	51
8.7.4	Example usage	51
8.7.5	SEE ALSO	51
8.8	HOBBIT-HOSTGRAPHS.CGI	52
8.8.1	SYNOPSIS	52
8.8.2	DESCRIPTION	52
8.8.3	OPTIONS	52
8.8.4	BUGS	52
8.8.5	SEE ALSO	52
8.9	BB-DATEPAGE.CGI	53
8.9.1	SYNOPSIS	53
8.9.2	DESCRIPTION	53
8.9.3	OPTIONS	53
8.9.4	ENVIRONMENT VARIABLES	53
8.9.5	SEE ALSO	54
8.10	BBDIGEST	55
8.10.1	SYNOPSIS	55
8.10.2	DESCRIPTION	55
8.10.3	EXAMPLE	55
8.10.4	SEE ALSO	55
8.11	HOBBITLAUNCH.CFG	56
8.11.1	SYNOPSIS	56
8.11.2	DESCRIPTION	56
8.11.3	FILE FORMAT	56
8.11.4	SEE ALSO	57
8.12	BB-EVENTLOG.CGI	58
8.12.1	SYNOPSIS	58
8.12.2	DESCRIPTION	58
8.12.3	OPTIONS	58
8.12.4	SEE ALSO	58
8.13	HOBBIT-MAILACK	59
8.13.1	SYNOPSIS	59
8.13.2	DESCRIPTION	59
8.13.3	ADDING INFORMATION TO THE REPLY MAIL	59
8.13.4	USE WITH PROCMAIL	59
8.13.5	USE WITH QMAIL	59
8.13.6	OPTIONS	59
8.13.7	SEE ALSO	59
8.14	BB-FINDHOST.CGI	60
8.14.1	SYNOPSIS	60
8.14.2	DESCRIPTION	60
8.14.3	OPTIONS	60
8.14.4	FILES	60
8.14.5	ENVIRONMENT VARIABLES	60
8.14.6	SEE ALSO	60
8.15	HOBBITCLIENT.CFG	61
8.15.1	DESCRIPTION	61

8.15.2 SETTINGS	61
8.15.3 INHERITED SETTINGS	61
8.15.4 SEE ALSO	61
9 Hobbit/BB Messages Protocols	62
9.1 BNF	62
9.1.1 Hobbit messages syntax	62
9.1.2 Big Brother messages syntax	63
9.2 bb - Hobbit client communication program	64
9.2.1 SYNOPSIS	64
9.2.2 DESCRIPTION	64
9.2.3 OPTIONS AND PARAMETERS	64
9.2.4 HOBBIT MESSAGE SYNTAX	64
9.2.5 EXAMPLE	67
9.2.6 NOTES	68
9.2.7 SEE ALSO	68
10 Data Backend	69
10.1 HOBBITD_RRD	69
10.1.1 SYNOPSIS	69
10.1.2 DESCRIPTION	69
10.1.3 OPTIONS	69
10.1.4 ENVIRONMENT	70
10.1.5 COLLECTED DATA	70
10.1.6 CUSTOM RRD DATA IN NAME-COLON-VALUE (NCV) FORMAT	71
10.1.7 CUSTOM RRD DATA VIA SCRIPTS	72
10.1.8 COMPATIBILITY	73
10.1.9 SEE ALSO	74
11 Miscellaneous programs	75
11.1 HOBBITLAUNCH	76
11.1.1 SYNOPSIS	76
11.1.2 DESCRIPTION	76
11.1.3 OPTIONS	76
11.1.4 STARTING TASKS	76
11.1.5 SEE ALSO	77
A Making of this book	78
A.1 Using Mercurial's book template to build our bookor	78
A.2 On Windows	79
B Open Publication License	80
B.1 Requirements on both unmodified and modified versions	80
B.2 Copyright	80
B.3 Scope of license	80
B.4 Requirements on modified works	81
B.5 Good-practice recommendations	81
B.6 License options	81
Bibliography	82
Index	82

List of Figures

List of Tables

3.1	Monitored Network Services	15
3.2	Monitored Network Services with SSL enabled	15
3.3	Other Monitored Network Services	15

Chapter 1

Preface

This book is all about hobbitmon developement work.

- RTFM: Read The Fine Manpage is the most direct and efficient way of learning a Unix subject. But it is hard if you trying to read a set of manpages.
- when man pages about Hobbit is 58 from author Henry.
- Problems of this approach is that manpage is not a book. it has no Table of Content,Indexes to locate a term easily.

1.1 This book is a work in progress

I am writing this book about hobbit from perspective of manpages.

I am releasing this Hobbit RTFM book while I am still writing it, in the hope that it will prove useful to others. I also hope that readers will contribute as they see fit.

1.2 Hobbit Documentation Road Map

- Hobbit RTFM.
- Hobbit User Guide.
- Hobbit Developer Guide.
- Hobbit Administration Guide.
- Hobbit Slides.

1.3 Revision History

- Henrik Storner
 - 1. Wrote the original manpages in troff format.
- T.J. Yang:
 - 1. Import from troff source files to LaTeX format.
 - 2. Fix LaTeX file into a chapter base tex syntax.

1.4 Colophon—this book is Free

This book is licensed under the Open Publication License, and is produced entirely using Free Software tools. It is typeset with \LaTeX ; illustrations are drawn and rendered with **Inkscape**.

The complete source code for this book is published as a SVN repository, at <http://hobbitmon.svn.sourceforge.net/viewvc/hobbitmon/branches/tjyang/src/books/hobbitmon-rtfm/en/>.

Chapter 2

Hobbit Overview

2.1 Hobbit - Introduction to the Hobbit Monitor

[Wau] Hobbit is a tool for monitoring the health of your networked servers and the applications running on them. It provides a simple, intuitive way of checking the health of your systems from a webbrowser, and can also alert you to any problems that arise through alarms sent as e-mail, SMS messages, via a pager or by other means.

Hobbit is Open Source software, licensed under the GNU GPL. This means that you are free to use Hobbit as much as you like, and you are free to re-distribute it and change it to suit your specific needs. However, if you change it then you must make your changes available to others on the same terms that you received Hobbit originally. See the file COPYING in the Hobbit source-archive for details.

Hobbit initially began life as an enhancement to Big Brother called “bbgen”. Over a period of 5 years, Hobbit has evolved from a small add-on to a full-fledged monitoring system with capabilities far exceeding what was in the original Big Brother package. Hobbit does still maintain some compatibility with Big Brother, so it is possible to migrate from Big Brother to Hobbit without too much trouble.

Migrating to Hobbit will give you a significant performance boost, and provide you with much more advanced monitoring. The Hobbit tools are designed for installations that need to monitor a large number of hosts, with very little overhead on the monitoring server. Monitoring of thousands of hosts with a single Hobbit server is possible - it was developed to handle just this task.

2.2 FEATURES

These are some of the core features in Hobbit:

Monitoring of hosts and networks Hobbit collects information about your systems in two ways: From querying network services (Web, LDAP, DNS, Mail etc.), or from scripts that run either on the Hobbit server or on the systems you monitor. The Hobbit package includes a Hobbit client which you can install on the servers you monitor; it collects data about the CPU-load, disk- and memory-utilisation, logfiles, network ports in use, file- and directory-information and more. All of the information is stored inside Hobbit, and you can define conditions that result in alerts, e.g. if a network service stops responding, or a disk fills up.

Centralized configuration All configuration of Hobbit is done on the Hobbit server. Even when monitoring hundreds or thousands of hosts, you can control their configuration centrally on the Hobbit server - so there is no need for you to login to a system just to change e.g. which processes are monitored.

Works on all major platforms The Hobbit server works on all Unix-like systems, including Linux, Solaris, FreeBSD, AIX, HP-UX and others. The Hobbit client supports all major Unix platforms, and there are other Open Source projects - e.g. BBWin, see <http://bbwin.sourceforge.net/> - providing support for Microsoft Windows based systems.

A simple, intuitive web-based front-end “Green is good, red is bad”. Using the Hobbit webpages is as simple as that. The hosts you monitor can be grouped together in a way that makes sense in your organisation and presented in a tree-structure. The webpages use many techniques to convey information about the monitored systems, e.g. different icons can be used for recently changed statuses; links to subpages can be listed in multiple columns; different icons can be used for dialup-tests or reverse-tests; selected columns can be dropped or unconditionally included on the webpages to eliminate unwanted information, or always include certain information; user-friendly names can be shown for hosts regardless of their true hostname. You can also have automatic links to on-line documentation, so information about your critical systems is just a click away.

Integrated trend analysis, historical data and SLA reporting Hobbit stores trend- and availability-information about everything it monitors. So if you need to look at how your systems behave over time, Hobbit has all of the information you need: Whether is response times of your webpages during peak hours, the CPU utilisation over the past 4 weeks, or what the availability of a site was compared to the SLA - it's all there inside Hobbit. All measurements are tracked and made available in time-based graphs.

When you need to drill down into events that have occurred, Hobbit provides a powerful tool for viewing the event history for each statuslog, with overviews of when problems have occurred during the past and easy-to-use zoom-in on the event.

For SLA reporting, You can configure planned downtime, agreed service availability level, service availability time and have Hobbit generate availability reports directly showing the actual availability measured against the agreed SLA. Such reports of service availability can be generated on-the-fly, or pre-generated e.g. for monthly reporting.

Role-based views You can have multiple different views of the same hosts for different parts of the organisation, e.g. one view for the hardware group, and another view for the webmasters - all of them fed by the same test tools.

If you have a dedicated Network Operations Centre, you can configure precisely which alerts will appear on their monitors - e.g. a simple anomaly in the system logfile need not trigger a call to 3rd-level support at 2 AM, but if the on-line shop goes down you do want someone to respond immediately. So you put the webcheck for the on-line shop on the NOC monitor, and leave out the log-file check.

Also for the techies The Hobbit user-interface is simple, but engineers will also find lots of relevant information. E.g. the data that clients report to Hobbit contain the raw output from a number of system commands. That information is available directly in Hobbit, so an administrator no longer needs to login to a server to get an overview of how it is behaving - the very commands they would normally run have already been performed, and the results are on-line in Hobbit.

Easy to adapt to your needs Hobbit includes a lot of tests in the core package, but there will always be something specific to your setup that you would like to watch. Hobbit allows you to write test scripts in your favourite scripting language and have the results show up as regular status columns in Hobbit. You can trigger alerts from these, and even track trends in graphs just by a simple configuration setting.

Real network service tests The network test tool knows how to test most commonly used protocols, including HTTP, SMTP (e-mail), DNS, LDAP (directory services), and many more. When checking websites, it is possible to not only check that the webserver is responding, but also that the response looks correct by matching the response against a pre-defined pattern or a checksum. So you can test that a network service is really working and supplying the data you expect - not just that the service is running.

Protocols that use SSL encryption such as https-websites are fully supported, and while checking such services the network tester will automatically run a check of the validity of the SSL server certificate, and warn about certificates that are about to expire.

Highly configurable alerts You want to know when something breaks. But you don't want to get flooded with alerts all the time. Hobbit lets you define several criteria for when to send out an alert, so you only get alerts when there is really something that needs your attention right away. While you are handling an incident, you can tell Hobbit about it so it stops sending more alerts, and so that everyone else can check with Hobbit and know that the problem is being taken care of.

Combined super-tests and test interdependencies If a single test is not enough, combination tests can be defined that combine the result of several tests to a single status-report. So if you need to monitor that at least 3 out of 5 servers are running at any time, Hobbit can do that for you and generate the necessary availability report.

Tests can also be configured to depend on each other, so that when a critical router goes down you will get alerts only for the router - and not from the 200 hosts behind the router.

2.3 SECURITY

All of the Hobbit server tools run under an unprivileged user account. A single program - the *hobbitping(1)* network connectivity tester - must be installed setuid-root, but has been written so that it drops all root privileges immediately after performing the operation that requires root privileges.

It is recommended that you setup a dedicated account for Hobbit.

Communications between the Hobbit server and Hobbit clients use the Big Brother TCP port 1984. If the Hobbit server is located behind a firewall, it must allow for inbound connections to the Hobbit server on tcp port 1984. Normally, Hobbit clients - i.e. the servers you are monitoring - must be permitted to connect to the Hobbit server on this port. However, if that is not possible due to firewall policies, then Hobbit includes the *hobbitfetch(8)* and *msgcache(8)* tools to allows for a pull-style way of collecting data, where it is the Hobbit server that initiates connections to the clients.

The Hobbit webpages are dynamically generated through CGI programs.

Access to the Hobbit webpages is controlled through your webserver access controls, e.g. you can require a login through some form of HTTP authentication.

2.4 DEMONSTRATION SITE

A site running this software can be seen at <http://www.hswn.dk/hobbit/>

2.5 PREREQUISITES

You will need a Unix-like system (Linux, Solaris, HP-UX, AIX, FreeBSD, Mac OS X or similar) with a webserver installed. You will also need a C compiler and some additional libraries, but many systems come with the required development tools and libraries pre-installed. The required libraries are:

1. **RRDtool** This library is used to store and present trend-data. It is required.
2. **libpcre** This library is used for advanced pattern-matching of text strings in configuration files. This library is required.
3. **OpenSSL** This library is used for communication with SSL-enabled network services. Although optional, it is recommended that you install this for Hobbit since many network tests do use SSL.
4. **OpenLDAP** This library is used for testing LDAP servers. Use of this is optional.

2.6 INSTALLATION

For more detailed information about Hobbit system requirements and how to install Hobbit, refer to the online documentation “Installing Hobbit” available from the Hobbit webserver (via the “Help” menu), or from the “docs/install.html” file in the Hobbit source archive.

2.7 SUPPORT and MAILING LISTS

hobbit@hswn.dk is an open mailing list for discussions about Hobbit. If you would like to participate, send an e-mail to **hobbit-subscribe@hswn.dk** to join the list.

An archive of the mailing list is available at <http://www.hswn.dk/hobbiton/>

If you just want to be notified of new releases of Hobbit, please subscribe to the hobbit-announce mailing list. This is a moderated list, used only for announcing new Hobbit releases. To be added to the list, send an e-mail to **hobbit-announce-subscribe@hswn.dk**.

2.8 Hobbit Wiki Book

[Yan] System Monitoring with Hobbit wiki book.

It is a community effort for hobbit documentation.

1. User Guide http://en.wikibooks.org/wiki/System_Monitoring_with_Hobbit/User_Guide
2. User Guide Administration Guide http://en.wikibooks.org/wiki/System_Monitoring_with_Hobbit/Administration_Guide
3. Developer Guide http://en.wikibooks.org/wiki/System_Monitoring_with_Hobbit/Developer_Guide
4. Other Docs http://en.wikibooks.org/wiki/System_Monitoring_with_Hobbit/Other_Docs

2.9 HOBBIT SERVER DAEMONS

These daemons implement the core functionality of the Hobbit server:

1. *hobbitd(8)* is the core daemon that collects all reports about the status of your hosts. It uses a number of helper modules to implement certain tasks such as updating logfiles and sending out alerts: *hobbitd_client*, *hobbitd_history*, *hobbitd_alert* and *hobbitd_rrd*. There is also a *hobbitd_filestore* module for compatibility with Big Brother.
2. *hobbitd_channel(8)* Implements the communication between the Hobbit daemon and the other Hobbit server modules.
3. *hobbitd_history(8)* Stores historical data about the things that Hobbit monitors.
4. *hobbitd_rrd(8)* Stores trend data, which is used to generate graphs of the data monitored by Hobbit.
5. *hobbitd_alert(8)* handles alerts. When a status changes to a critical state, this module decides if an alert should be sent out, and to whom.
6. *hobbitd_client(8)* handles data collected by the Hobbit clients, analyzes the data and feeds back several status updates to Hobbit to build the view of the client status.
7. *hobbitd_hostdata(8)* stores historical client data when something breaks. E.g. when a webpage stops responding *hobbitd_hostdata* will save the latest client data, so that you can use this to view a snapshot of how the system state was just prior to it failing.

2.10 HOBBIT NETWORK TEST TOOLS

These tools are used on servers that execute tests of network services.

1. *hobbitping(1)* performs network connectivity (ping) tests.
2. *bbtest-net(1)* runs the network service tests.
3. *bbretest-net.sh(1)* is an extension script for re-doing failed network tests with a higher frequency than the normal network tests. This allows Hobbit to pick up the recovery of a network service as soon as it happens, resulting in less downtime being recorded.

2.11 HOBBIT TOOLS HANDLING THE WEB USER-INTERFACE

These tools take care of generating and updating the various Hobbit web-pages.

1. *bbgen(1)* takes care of updating the Hobbit webpages.
2. *hobbitsvc.cgi(1)* This CGI program generates an HTML view of a single status log. It is used to present the Hobbit status-logs.
3. *hobbitgraph.cgi(1)* This CGI program generates graphs of the trend-data collected by Hobbit.
4. *hobbit-hostgraphs.cgi(1)* When you want to combine multiple graphs into one, this CGI lets you combine graphs so you can e.g. compare the load on all of the nodes in your server farm.
5. *hobbit-nkview.cgi(1)* Generates the Critical Systems view, based on the currently critical systems and the configuration of what systems and services you want to monitor when.
6. *bb-hist.cgi(1)* This CGI program generates a webpage with the most recent history of a particular host+service combination.
7. *bb-eventlog.cgi(1)* This CGI lets you view a log of events that have happened over a period of time, for a single host or test, or for multiple systems.
8. *bb-ack.cgi(1)* This CGI program allows a user to acknowledge an alert he received from Hobbit about a host that is in a critical state. Acknowledging an alert serves two purposes: First, it stops more alerts from being sent so the technicians are not bothered with more alerts, and secondly it provides feedback to those looking at the Hobbit webpages that the problem is being handled.
9. *hobbit-mailack(8)* is a tool for processing acknowledgements sent via e-mail, e.g. as a response to an e-mail alert.
10. *hobbit-enadis.cgi(8)* is a CGI program to disable or re-enable hosts or individual tests. When disabling a host or test, you stop alarms from being sent and also any outages do not affect the SLA calculations. So this tool is useful when systems are being brought down for maintenance.
11. *bb-findhost.cgi(1)* is a CGI program that finds a given host in the Hobbit webpages. As your Hobbit installation grows, it can become difficult to remember exactly which page a host is on; this CGI script lets you find hosts easily.
12. *bb-rep.cgi(1)* This CGI program triggers the generation of Hobbit availability reports, using *bbgen(1)* as the reporting back-end engine.
13. *bb-replog.cgi(1)* This CGI program generates the detailed availability report for a particular host+service combination.

14. *bb-snapshot.cgi(1)* is a CGI program to build the Hobbit webpages in a “snapshot” mode, showing the look of the webpages at a particular point in time. It uses *bbgen(1)* as the back-end engine.
15. *hobbit-statusreport.cgi(1)* is a CGI program reporting test results for a single status but for several hosts. It is used to e.g. see which SSL certificates are about to expire, across all of the Hobbit webpages.
16. *bb-csvinfo.cgi(1)* is a CGI program to present information about a host. The information is pulled from a CSV (Comma Separated Values) file, which is easily exported from any spreadsheet or database program.

2.12 CLIENT-SIDE TOOLS

1. *logfetch(1)* is a utility used by the Hobbit Unix client to collect information from logfiles on the client. It can also monitor various other file-related data, e.g. file metadata or directory sizes.
2. *clientupdate(1)* Is used on Hobbit clients, to automatically update the client software with new versions. Through this tool, updates of the client software can happen without an administrator having to logon to the server.
3. *msgcache(8)* This tool acts as a mini Hobbit server to the client. It stores client data internally, so that the *hobbitfetch(8)* utility can pick it up later and send it to the Hobbit server. It is typically used on hosts that cannot contact the Hobbit server directly due to network- or firewall-restrictions.

2.13 HOBBIT COMMUNICATION TOOLS

These tools are used for communications between the Hobbit server and the Hobbit clients. If there are no firewalls then they are not needed, but it may be necessary due to network or firewall issues to make use of them.

1. *bbproxy(8)* is a proxy-server that forwards Hobbit messages between clients and the Hobbit server. The clients must be able to talk to the proxy, and the proxy must be able to talk to the Hobbit server.
2. *hobbitfetch(8)* is used when the client is not able to make outbound connections to neither bbproxy nor the Hobbit server (typically, for clients located in a DMZ network zone). Together with the *msgcache(8)* utility running on the client, the Hobbit server can contact the clients and pick up their data.

2.14 OTHER TOOLS

1. *hobbitlaunch(8)* is a program scheduler for Hobbit. It acts as a master program for running all of the Hobbit tools on a system. On the Hobbit server, it controls running all of the server tasks. On a Hobbit client, it periodically launches the client to collect data and send them to the Hobbit server.
2. *bb(1)* is the tool used to communicate with the Hobbit server. It is used to send status reports to the Hobbit server, through the custom Hobbit/BB protocol, or via HTTP. It can be used to query the state of tests on the central Hobbit server and retrieve Hobbit configuration files. The server-side script *bbmessage.cgi(1)* used to receive messages sent via HTTP is also included.
3. *bbcmd(1)* is a wrapper for the other Hobbit tools which sets up all of the environment variables used by Hobbit tools.
4. *bbhostgrep(1)* is a utility for use by Hobbit extension scripts. It allows an extension script to easily pick out the hosts that are relevant to a script, so it need not parse a huge *bb-hosts* file with lots of unwanted test-specifications.
5. *bbhostshow(1)* is a utility to dump the full *bb-hosts(5)* file following any “include” statements.
6. *bbdigest(1)* is a utility to compute message digest values for use in content checks that use digests.

7. *bbcombotest(1)* is an extension script for the Hobbit server, allowing you to build complicated tests from simpler Hobbit test results. E.g. you can define a test that uses the results from testing your webserver, database server and router to have a single test showing the availability of your enterprise web application.
8. *trimhistory(8)* is a tool to trim the Hobbit history logs. It will remove all log entries and optionally also the individual status-logs for events that happened before a given time.

2.15 VERSIONS

1. *Version 1* of bbgen was released in November 2002, and optimized the webpage generation on Big Brother servers.
2. *Version 2* of bbgen was released in April 2003, and added a tool for performing network tests.
3. *Version 3* of bbgen was released in September 2004, and eliminated the use of several external libraries for network tests, resulting in a significant performance improvement.
4. *Version 4.0* released on March 30 2005, the project was de-coupled from Big Brother, and the name changed to Hobbit. This version was the first full implementation of the Hobbit server, but it still used the data collected by Big Brother clients for monitoring host metrics.
5. *Version 4.1* was released in July 2005 included a simple Hobbit client for Unix. Logfile monitoring was not implemented.
6. *Version 4.2* was released in July 2006, and includes a fully functional Hobbit client for Unix.

2.16 COPYRIGHT

Hobbit is Copyright(C)2002-2007,HenrikStorner<henrik@storner.dk> Parts of the Hobbit sources are from public-domain or other freely available sources. These are the the Red-Black tree implementation, and the MD5-, SHA1-, SHA2- and RIPEMD160-implementations. Details of the license for these is in the README file included with the Hobbit sources. All other files are released under the GNU General Public License version 2, with the additional exemption that compiling, linking, and/or using OpenSSL is allowed. See the file COPYING for details.

2.17 SEE ALSO

1. *hobbitd(8)*
2. *hobbitd_channel(8)*
3. *hobbitd_history(8)*
4. *hobbitd_rrd(8)*
5. *hobbitd_alert(8)*
6. *hobbitd_client(8)*
7. *hobbitd_hostdata(8)*
8. *hobbitping(1)*
9. *bbtest-net(1)*
10. *bbretest-net.sh(1)*

11. `bbgen(1)`
12. `hobbitsvc.cgi(1)`
13. `hobbitgraph.cgi(1)`
14. `hobbit-hostgraphs.cgi(1)`
15. `hobbit-nkview.cgi(1)`
16. `bb-hist.cgi(1)`
17. `bb-eventlog.cgi(1)`
18. `bb-ack.cgi(1)`
19. `hobbit-mailack(8)`
20. `hobbit-enadis.cgi(8)`
21. `bb-findhost.cgi(1)`
22. `bb-rep.cgi(1)`
23. `bb-replog.cgi(1)`
24. `bb-snapshot.cgi(1)`
25. `hobbit-statusreport.cgi(1)`
26. `bb-csvinfo.cgi(1)`
27. `logfetch(1)`
28. `clientupdate(1)`
29. `msgcache(8)`
30. `bbproxy(8)`
31. `hobbitfetch(8)`
32. `hobbitlaunch(8)`
33. `bb(1)`
34. `bbmessage.cgi(1)`
35. `bbcnd(1)`
36. `bbhostgrep(1)`
37. `bbhostshow(1)`
38. `bbdigest(1)`
39. `bbcombotest(1)`
40. `trimhistory(8)`
41. `bb-hosts(5)`
42. `hobbitlaunch.cfg(5)`

- 43. `hobbitserver.cfg(5)`
- 44. `hobbit-alerts.cfg(5)`
- 45. `hobbit-clients.cfg(5)`
- 46. `client-local.cfg(5)`

Chapter 3

Configuring Hobbit Monitoring

3.1 Configuring Hobbit Monitoring

The Hobbit configuration is kept in the files in the `/server/etc/` directory. If you look at this directory, you will see these files:

- “`/etc/opt/hobbitserver42/bb-hosts`” is the one you will change the most. This file contains a list of all the hosts you are monitoring, including information such as their IP-address, what network services you are monitoring on the host, what URL’s you are checking, what subpage in the Hobbit web-pages this host is shown on etc. The name of the file - “bb-hosts” - was chosen so it is compatible with the Big Brother system which uses the same filename and file format.
- “`/etc/opt/hobbitserver42/hobbit-clients.cfg`” is the configuration file for data reported by the Hobbit clients installed on the hosts you are monitoring. This defines the color of the cpu-, disk-, memory- and procs- columns, based on the information that is sent to Hobbit by the clients.
- “`/etc/opt/hobbitserver42/hobbit-alerts.cfg`” holds the alerting configuration. In this file, you setup the rules for sending out alerts about services going down: Who gets the alert, how is it sent, how often, whether to send alerts 24x7 or only between 10 AM and 4 PM on weekdays etc.
- “`/etc/opt/hobbitserver42/hobbitserver.cfg`” is the configuration file for the Hobbit server. This file defines a lot of environment variables that are made available to all of the Hobbit programs when they run. Some environment variables that are defined in the Big Brother system are also setup by Hobbit, so that Big Brother extension scripts will work.

The initial configuration of `hobbitserver.cfg` is setup by the `configure` script when you install Hobbit, and in most cases you will not need to change it.

- “`/etc/opt/hobbitserver42/hobbitlaunch.cfg`” is the configuration file for the `hobbitlaunch` tool. `hobbitlaunch` is the master program in Hobbit, it is the only program you start to run the Hobbit server. `hobbitlaunch` reads the `hobbitlaunch.cfg` file, and starts the programs listed here to run the server. Some of the programs may run as daemons, some of the programs may run at regular intervals. If you want to use some of the advanced options for the `bbgen` or `bbtest-net` programs, you change the `hobbitlaunch.cfg` file to add these options to the commandline.
- “`/etc/opt/hobbitserver42/hobbitgraph.cfg`” is a configuration file for the `hobbitgraph` CGI. It defines how the graphs are generated from the data in the RRD files.
- `bb-services` is a configuration file for the `bbtest-net` program. It defines how network services are checked.

3.2 Setting up monitoring of hosts

The bb-hosts file defines which hosts Hobbit monitors. When you install Hobbit, a simple configuration is setup that just lists the Hobbit server:



```

Session Redigér Vis Bogmærker Opsætning Hjælp
-bash-2.05b$ cd ~/server/etc/
-bash-2.05b$ cat bb-hosts
#
# Master configuration file for Hobbit and Big Brother
#
# This file defines several things:
#
# 1) By adding hosts to this file, you define hosts that are monitored
#    by Hobbit / Big Brother.
# 2) By adding "page", "subpage", "group" definitions, you define the layout
#    of the Hobbit / Big Brother webpages, and how hosts are divided among
#    the various webpages that Hobbit / Big Brother generates.
# 3) Several other definitions can be done for each host, see the bb-hosts(5)
#    man-page.
#
# You need to define at least the Hobbit server itself here.

80.62.63.88 www.hswn.dk      # BBDISPLAY BBPAGER BBNET bbd http://www.hswn.dk/

-bash-2.05b$ █

```

There are a few things to notice here:

- Lines that begin with a # are comments.
- Each host you monitor is on a line by itself, with the IP-address and the hostname of the host.
- You can add extra tags to each host definition, by putting in a #-mark and then some keywords. These keywords define how Hobbit handles the host.

The bb-hosts file shown in the example has only one host defined: `www.hswn.dk` which is the server running Hobbit. There are a few extra keywords thrown in:

- `BBDISPLAY`, `BBPAGER`, `BBNET` are compatibility settings for extensions written for Big Brother. Hobbit doesn't use these, but puts them in the bb-hosts file to avoid problems if you mix Hobbit and Big Brother modules.
- `bbd` is the name of a *network test*. This keyword causes the Hobbit network-tester `bbtest-net` to check if the `bbd` network service is running on this host, and send a status report to the Hobbit server with the information about this service. So you'll get a (hopefully!) green icon on the Hobbit webpage for this host, showing the status of the `bbd` network service.

Network services are defined in the `bb-services` file, so this file must have an entry for `bbd` defining what TCP port to check, and possibly also what data to send to the service and what to expect as a response.

- `http://www.hswn.dk/` is a URL, obviously. This also triggers a network test, the Hobbit network tester will try to request this URL, and send in a status report showing if the URL was accessible or not.

By default, Hobbit will always check if the host is up and running by trying to “ping” it. This results in a `conn` column on the Hobbit webpage for this host, showing if the ping-test succeeded. If you have a host that does not respond to ping - e.g. because there is a firewall that filters out such requests - then you can disable the ping-test by putting a “noconn” keyword on the line in bb-hosts.

As you can see, the syntax is pretty straight-forward. Need to monitor an extra URL for this server ? Just add the URL to the line. Need to check if ssh (Secure Shell) is running ? Just add `ssh`

Table 3.1: Monitored Network Services

conn	Simple ping test. Enabled by default, you can disable it by putting “noconn” into bb-hosts.
http	Web-server test. Enter the URL to request from the webserver.
ftp	FTP server test.
ssh	SSH (Secure Shell) server test. Supports ssh1 and ssh2.
telnet	Telnet server test.
smtp	SMTP (Mail server) test.
pop3	POP-3 test.
imap	IMAP test. IMAP version 2 and 4 are supported, for version 3 use “imap3”.
nntp	NNTP (News) server test.
ldap	LDAP (Directory server) test. Enter the full LDAP URI if Hobbit is configured with LDAP support.
rsync	rsync server test.
bdb	Big Brother daemon test. Also works with the Hobbit network daemon.
clamd	CLAM anti-virus daemon test.
spamd	SpamAssassin anti-spam daemon test.
oratns	Oracle TNS listener test. Will attempt to do an oratns “ping”.
qsmtp	QSMTP server test. For qmail’s qmtpd service.
qmqp	QMQP server test. For qmail’s mqpd service.

Table 3.2: Monitored Network Services with SSL enabled

https	Web-server test. Enter the URL to request from the webserver.
ftps	Secure FTP server test.
telnets	Secure Telnet server test.
smtps	Secure SMTP server test.
pop3s	Secure POP-3 server test.
imaps	Secure IMAP server test.
nntps	Secure NNTP (News) server test.
ldaps	Secure LDAP (Directory) server test.

to the line. The full set of keywords you can use is described in the bb-hosts man-page. Many of the keywords relate to the way Hobbit displays the information about the host on the web-pages, other keywords deal with how the uptime percentage is calculated for availability reports, and some keywords - like the *bdb* and *http://...* mentioned above - describe the network services that are tested for this host.

Monitoring network services

As shown in the example above, adding a network test for a host is as simple as putting the right keyword into the bb-hosts file. The default set of network tests configured in Hobbit 4.0 is as follows:

If Hobbit is built with OpenSSL support, the following SSL-enabled services can also be checked:

Enter the full LDAP URI if Hobbit is configured with LDAP support. Note that this is only possible when Hobbit is built with the OpenLDAP v2.x client library, and only for LDAP servers that support LDAP version 3 and the “starttls” command. LDAP server that use the older non-standard method of tunnelling LDAP through SSL on port 636 will not work.

There are a few network tests that Hobbit can run for you, by using external programs. This is not a very effective way of testing, so it is only done this way for a few very specialised tests:

Table 3.3: Other Monitored Network Services

ntp	NTP (Network Time protocol) server test, using the “ntpd” command.
rpc	RPC service test. This queries the <i>portmapper</i> service on the server, using the “rpcinfo” command. See the bb-hosts(5) man-page.

Monitoring host-specific data with clients

You can install a client on each of the hosts you monitor, to check host-specific data such as CPU utilisation, disk usage, if certain processes and services are running etc. Hobbit includes clients for most Unix-like operating systems. A client for Windows is planned but the programming has not yet started.

First, make sure you have installed the Hobbit client on all of the hosts you want to monitor, and you have these hosts listed in your bb-hosts file. The Hobbit client will pick up the hostname of the box it is running on automatically, but it is not uncommon for the name it finds to be different from what you've put into bb-hosts. So if you know that the client is running but no data appears, check that the hostname used by the Hobbit client is the one you expect. See this FAQ item for details.

With the Hobbit client running and reporting data into Hobbit, you should see the cpu-, disk-, memory- and procs- columns appear. The color of these status columns is determined by settings in the `hobbit-clients.cfg` configuration file. Here is an example of how to setup a host:



The screenshot shows a window titled "Session" with a menu bar including "Redigér", "Vis", "Bogmærker", "Opsætning", and "Hjælp". The main area displays host-specific data for "voodoo.hswn.dk". The data is organized into sections: "HOST=voodoo.hswn.dk", "UP 30m", "LOAD 6.0 12.0", "DISK /export/netdisk 93 98", "DISK /export/home 85 95", "DISK * 70 85", and "PROC" entries for cron, apache2, svscan, hobbitd_history, hobbitd_rrd, sshd, dnscache, dhcpcd, hobbitd, and hobbitlaunch. Below the main window are icons for "Skal" and a "Help" button.

```

HOST=voodoo.hswn.dk
  UP 30m
  LOAD 6.0 12.0
  DISK /export/netdisk 93 98
  DISK /export/home 85 95
  DISK * 70 85
  PROC cron 1 -1 yellow
  PROC apache2 1 -1 yellow
  PROC svscan 1 -1 yellow
  PROC *^hobbitd_history 1 -1 yellow
  PROC *^hobbitd_rrd 2 -1 yellow
  PROC sshd
  PROC dnscache
  PROC dhcpcd
  PROC hobbitd
  PROC hobbitlaunch

```

As you can see, there's first a definition of what hosts the following criteria applies to. Here, it is only a single host: `voodoo.hswn.dk` - but you can use various filters on hostnames, pagenames and time of day to determine what the thresholds should be for each of the criteria monitored with the client data. The `hobbit-clients.cfg` man-page describes this in detail.

After the host filter comes the criteria used to determine the color of each of the status columns.

UP

LOAD

column color, based on how much load is on the system. After the LOAD keyword you put two limits: The first number is the limit

DISK

The name of the filesystem is the mount point. You can specify this either with the full path, or you can use * meaning “all filesystems”

PROC

MEMSWAP

LOG

More about logfile monitoring

Configuring the LOG entries in the `hobbit-clients.cfg` file is only one half of the configuration - you also need to tell the Hobbit client running on the monitored system that it must send in some data from that logfile in the first place. For that, you must configure the `client-local.cfg` file with the name of the logfile.

Upgrading from Big Brother to Hobbit

3.3 Upgrading from Big Brother to Hobbit

First, you should realize that this is not a fully automated process. You will need to do some work yourself - especially with the handling of alerts.

First step: Install Hobbit

To begin, install Hobbit as described in the Hobbit installation guide. I recommend that you configure Hobbit to use the same user-ID as your current Big Brother installation, but have it use a different directory for the server- and data-files. The default is to use */server* and */data* respectively, which is unlikely to clash with the directories where you have Big Brother installed. If you do need to change the directories, you must edit the top-level Makefile and change the BBHOME and BBVAR settings near the top of the file.

Step two: Move the configuration files

A couple of configuration files can be copied directly from Big Brother to Hobbit:

- The **bb-hosts** file
- The **bb-services** file. You need only copy this if you have used bbgen before, and added custom network tests to the bb-services file.
- The **cookies** file. You may not have this file - it is only present if you have used bbgen before and have setup HTTP tests that require cookies.
- The **bbcombotests.cfg** file. You may not have this file - it is only present if you have used bbgen before and have setup the bbcombotest tool.

The *bbwarnrules.cfg* and *bbwarnsetup.cfg* files cannot be copied over. Hobbit uses a very different configuration file for the alert configuration, so you will have to re-write your alert configuration for Hobbit. See the Hobbit alert configuration to learn how Hobbit alerts are configured.

Any server-side extension-scripts can be copied from the \$BBHOME/ext/ directory to the /server/ext/ directory. You must also add entries for them to the Hobbit *hobbitlaunch.cfg* file. Beware that many scripts rely on environment variables that Big Brother defines, but which Hobbit does not define - in that case, you need to setup those environment variables in the *hobbitserver.cfg* file. It is probably easiest to save this until you start running Hobbit, and can look at any error-output from the scripts.

If you have modified the webpage header- and footer-files in \$BBHOME/web/ then you can copy the modified files over directly to the /server/web/ directory. Note that Hobbit has a number of header- and footer-files for the various CGI scripts that are not present in Big Brother, so you may need to setup a few extra files to get a consistent look across your new Hobbit installation.

Step three: Stop Big Brother

You are now going to move over the data files. To avoid confusion about files being updated by Big Brother while they are being moved over to Hobbit, I recommend that you stop Big Brother now.

Step four: Move the history logs

You may want to save the historical logfiles and the history of your status changes. To do that, move all of the files or directories in the \$BBVAR/hist/ to the /data/hist/ directory, and all of the files or directories in \$BBVAR/histlogs/ to the /data/histlogs/ directory. If you prefer to keep them in the Big Brother directory, you can copy them over with “cp -r” or “tar” instead of moving them.

Step five: Move the RRD files

The RRD files are used to generate the graphs, if you have installed the LARRD add-on to Big Brother. Hobbit has RRD support built-in, and it is obviously nice to keep the historical data that has been collected over time.

The filesystem layout of the RRD files is different from Big Brother+LARRD to Hobbit. Instead of having all of the RRD files in one big directory, there is a subdirectory for each host holding only the RRD files for data from that host. This is easier to manage, and also speeds up the graph generation when you have many hosts. Unfortunately, it makes migrating from Big Brother to Hobbit slightly more complicated.

In the Hobbit source-tree, you will find a script *hobbitd/moverrd.sh*. This script moves or copies the RRD files from the Big Brother+LARRD structure into the Hobbit structure. You must edit a couple of settings at the beginning of the file, especially to set the correct directory where Big Brother stores your current RRD files (the SRCDIR setting). By default the script *copies* the files over to the new structure, if you would rather just move them then change to “OP” setting to “mv”.

After setting up the script, run it and it should copy all of the RRD-files that relate to a host currently in the bb-hosts file to the new directory structure.

Step 6: Start Hobbit

Start Hobbit with the */server/hobbit.sh start* command. Look at the logfiles in the */var/log/hobbit* directory (or elsewhere, if you did not choose the default logfile directory when configuring Hobbit) and fix any problems that show up.

Look at the webpages generated. For the first few minutes, there will be some missing columns and icons for each host, since it takes some time for all of the tests to report a status to the new Hobbit daemon. After 5-10 minutes all of the standard tests should appear.

Chapter 4

Integrating MRTG data into Hobbit

4.1 Integrating MRTG data into Hobbit

This document describes one way of integrating MRTG graphs into Hobbit. It's simple, doesn't require any additional scripts, and provides all of your MRTG graphs as part of the "trends" column that is already present for all hosts in Hobbit.

Another way of doing this is the bb-mrtg.pl script. This is an extension script that gives you some more options for controlling where the graphs show up, and also lets you generate alerts based on data collected by MRTG.

Simple Hobbit-MRTG support

MRTG by default uses its own fileformat for the data files, and continuously generates PNG- or GIF-images of the data. This is a waste of ressources - most of the time, these images are never seen. This was in fact one of the reasons that RRDtool was developed, to separate the data-collection from the graph generation.

Hobbit uses the RRDtool format for all of its data. You can configure MRTG to save data using the RRDtool data format, instead of the default MRTG log-file format. This lets your MRTG save the data directly into the Hobbit RRD directory, in the same format that all of the other Hobbit RRD files use. You can then use the normal Hobbit graph tools to view the graphs.

To configure MRTG to use the RRDtool format, you must setup the mrtg.cfg file like this at the top of the file:

```
# For Hobbit integration
WorkDir: /usr/local/hobbit/data/rrd
LogFormat: rrdtool
```

Note that the WorkDir setting points to the top-level RRD directory, i.e. the one defined via the BBRRDS setting in hobbitserver.cfg. The Logformat:rrdtool makes MRTG save data using the RRDtool data format.

Each of the network interfaces you monitor have a target-definition in the mrtg.cfg file. You need to modify this slightly, to make it save the RRD data file in a subdirectory matching the hostname you have in the bb-hosts file, and with a filename that begins with "mrtg.". Like this:

```
Target[mrtg.myrouter.eth0]: /10.0.0.1:public@myrouter.sample.com:
Directory[mrtg.myrouter.eth0]: myrouter.sample.com
```

This defines an MRTG target, where it monitors the interface on **myrouter.sample.com** that has the IP-address **10.0.0.1**. It uses the community name **public** to query the SNMP daemon on the router.

The **Directory[mrtg.myrouter.eth0]:myrouter.sample.com** instructs MRTG to save the data file in this directory relative to the **WorkDir** directory, i.e. the final directory for the RRD datafile will be /usr/local/hobbit/data/rrd/myrouter.sample.com which is where Hobbit expects all of the RRD-files for the myrouter.sample.com host to be. The name of the RRD data-file will be **mrtg.myrouter.eth0.rrd** - i.e. the name of the target.

The reason for naming the data file **mrtg.*.rrd** is that the hobbitgraph tool has a built-in definition for generating graphs from this type of files. So if you stick to this naming convention, the graphs will automatically show up on the Hobbit “trends” page. If you have more than one device that you collect data from, you’ll need to modify this; you can use any name for the target as long as it is of the form **mrtg.*.DEVICE** - i.e. first “mrtg.”, then some random text (e.g. the hostname), then a dot and the device-name. The device-name is used as a legend on the graphs, so you probably want to make this something recognizable, like the name of the network interface, or some sensible description like “DSL”, “LAN”, “T1” or whatever you know your devices as. Note the MRTG converts this to lower-case.

Here is the full mrtg.cfg configuration used to track traffic on my Internet gateway (currently a 4 Mbit/512 Kbit ADSL). Note that even though MRTG does not use the Title and MaxBytes settings, they are required - MRTG will not run without them:

```
# For Hobbit integration
WorkDir: /var/lib/hobbit/rrd
LogFormat: rrdtool

# The external interface on my router
Directory[mrtg.fenris.dsl]: fenris.hswndk
Target[mrtg.fenris.dsl]: /80.62.63.88:public@fenris:
Title[mrtg.fenris.dsl]: Traffic Analysis for External DSL
MaxBytes1[mrtg.fenris.dsl]: 500000
MaxBytes2[mrtg.fenris.dsl]: 62500

# The internal interface on my router
Directory[mrtg.fenris.lan]: fenris.hswndk
Target[mrtg.fenris.lan]: /10.0.0.1:public@fenris:
Title[mrtg.fenris.lan]: Traffic Analysis for internal LAN
MaxBytes[mrtg.fenris.lan]: 1250000
```

With this setup, I have the MRTG graphs readily available on the “trends” page, together with all of the other Hobbit graphs.

Running the MRTG data collector from hobbitlaunch

Normally there is a cron job that runs the mrtg command every 5 minutes to collect the MRTG data. But you can run it from hobbitlaunch - this also has the benefit that the RRD files will be owned by the hobbit user.

All that is needed is to add a section for MRTG to Hobbit’s hobbitlaunch.cfg file. Mine looks like this:

```
[mrtg]
CMD /usr/bin/mrtg --lock-file $BBSERVERLOGS/mrtg.lock /etc/mrtg.cfg
INTERVAL 5m
LOGFILE $BBSERVERLOGS/mrtg.log
```

Some Linux distributions setup MRTG with the expectation that it will always be run by the root user. So you may have to change permissions on some files and directories e.g. to permit the hobbit user to read the mrtg.cfg file. Check the mrtg.log file for errors.

Chapter 5

Known issues in Hobbit

5.1 Known issues in Hobbit

This describes some known problems you may encounter when using Hobbit to monitor servers.

- How to report bugs
- JavaScript errors when using enable/disable function
- DNS error reported for network tests
- Network tests fail sporadically
- “Failed to find enough entropy” on Solaris
- Hobbit fails on FreeBSD with “Could not get sem: No space left on device”
- Hobbit will not run inside a FreeBSD jail
- Hobbit on Solaris compiles but aborts with some “ld.so” error
- Hobbit on Solaris compiles but aborts with some “ld.so” error
- Hobbit on Solaris 10 “hobbitd status-board not available”

How to report bugs

If you think you have found a bug in Hobbit, please report it to the Hobbit mailing list at hobbit@hswn.dk. You can do a lot to help getting bugs fixed by providing detailed information about the bug:

- Always include the version number of Hobbit you’re using
- If one of the Hobbit tools crashes and leaves a core-file (usually in the `hobbit/server/tmp/` directory), please use the `gdb` tool to pinpoint where the crash occurred:
 - Login as the Hobbit user
 - `$ cd /server`
 - `$ gdb bin/PROGRAMFILE tmp/COREFILE`
 - then at the `gdb>` prompt, execute the command
`gdb> bt`

Internet Explorer complains about Javascript errors in Enable/Disable

This happens for some, but works for most people. One workaround is to disable the Javascript validation code in the enable/disable function: Edit `hobbit/cgi-bin/hobbit-enadis.sh` script and add the option “`-no-jsvalidation`” to the `hobbisvc.cgi` command - this disables Javascript validation on the “info” page - and edit the file `hobbit/server/web/maint_form` so you remove the text `'onClick="validateDisable(this.form)'` from the input-tag near the end of that file.

DNS error reported for network tests

The bbtest-net network tester uses the built-in ARES library for DNS lookups. There have been reports that this library fails on some systems; one confirmed case is on “OSF1 V5.1”. So if you suddenly get a lot of failed network tests that report “DNS error”, try running bbtest-net with the “`-no-ares`” option to use the standard DNS lookups instead.

Network tests fail sporadically, or report long reponsetimes

The bbtest-net network tester runs many tests in parallel; by default it will typically run up to 256 tests concurrently. This may be more than your network test server or network infrastructure can handle; if you see sporadic timeouts of network tests or the graphs show increased responsetimes, you can lower the number of concurrent tests by adding the “`-concurrency=N`” option to bbtest-net in the `/server/etc/hobbitlaunch.cfg` file. This has been especially important for sites doing many http checks, since these typically have much more traffic going on while testing than simple TCP services such as smtp.

Network tests fail on Solaris with “Failed to find enough entropy”

OpenSSL uses random data to initialise a key that is negotiated when a new SSL-encrypted connection is being setup. Solaris 8 and earlier doesn’t have a standard way of getting random data, so OpenSSL cannot get all of the randomness it needs. Solaris **patch 112438** solves this by adding a `/dev/random` device that provides random data to applications. Thanks to Scott Kelley for the pointer to the Solaris patch.

Asif Iqbal notes: Patch 112438 only works for Solaris 8. For Solaris 6 and 7 you need to either install SUNWski pkg or ANDIrand pkg. See <http://www.cosy.sbg.ac.at/~andi/SUNrand/>. I have been using ANDIrand since that did not require a reboot and easily available.

Hobbit fails on FreeBSD with “Could not get sem: No space left on device”

Hobbit uses some kernel resources - semaphores and shared memory. If you get the following error message in `hobbitlaunch.log` when trying to start Hobbit:

```
2005-05-29 20:25:14 Setting up hobbitd channels
2005-05-29 20:25:14 Could not get sem: No space left on device
2005-05-29 20:25:14 Cannot setup status channel
2005-05-29 20:25:14 Task hobbitd terminated, status 1
```

then you need to increase the number of semaphore sets and individual semaphores available to applications. The current settings for your kernel can be found with “`sysctl kern.ipc.semmin`” (semaphore sets) and “`sysctl kern.ipc.semmaxns`” (total number of semaphores). Hobbit uses 6 semaphore sets, with a total of 18 semaphores.

To increase this, put these two lines in `/boot/loader.conf` on your system:

```
kern.ipc.semmin="40"
kern.ipc.semmaxns="100"
```

Adjust the values to something reasonable for your system - considering the current settings (from the sysctl output), and Hobbit's needs (6 sets with 18 semaphores).

More information about tuning the FreeBSD kernel parameters is available in the FreeBSD Handbook

Hobbit will not run inside a FreeBSD jail

FreeBSD jails have System V IPC support disabled by default. See the FreeBSD architecture Handbook section on jail restrictions. Essentially, there is a **security.jail.jail.sysvipc.allowed** sysctl that must be enabled. If it isn't then Hobbit will refuse to start with the error Could not get shm of size 262144: Function not implemented.

Hobbit on Solaris compiles but aborts with some “ld.so” error

This info was contributed by sladewig, with a few modifications:

The system loader/linker can't find your lib. Assuming you have the .so lib in /usr/local/lib, You can add -R to the Makefile PCRELIB

Note: This information only applies if you are using the Solaris linker. The GNU linker uses the “-rpath” option which is defined differently: Add

RPATH = -Wl,-rpath=

at the bottom of the top-level Makefile.

Hobbit on Solaris 10 “hobbitd status-board not available”

It seems that Hobbit triggers a bug in the Solaris 10 “TCP fusion” loopback interface, resulting in no data being returned from hobbitd when queried for the current status of all monitored systems. The recommended solution is to disable the use of TCP fusion. Colin Spargo writes:

If you disable TCP fusion be adding the following kernel parameter to /etc/system and reboot, hopefully you will find that the problem goes away.

```
set ip:do_tcp_fusion = 0
```

Apparently this can be done on a live system as well (without rebooting), but will require hobbit to be restarted. To do this:

```
echo do_tcp_fusion/W0 | mdb -kw
```

More in his mail

Chapter 6

Report Generation

6.1 HOBBIT-CONFREPORT.CGI

hobbit-confreport.cgi - Hobbit Configuration report

6.1.1 SYNOPSIS

hobbit-confreport.cgi

6.1.2 DESCRIPTION

hobbit-confreport.cgi is invoked as a CGI script via the `hobbit-confreport.sh` CGI wrapper.

hobbit-confreport.cgi provides a plain HTML (Web) report of the Hobbit configuration for a group of hosts; which hosts are included is determined by the hosts available on the webpage from where the CGI script is invoked.

The configuration report include the hostnames, a list of the statuses monitored for each host, and if applicable any configuration settings affecting these. Alerts that may be triggered by status changes are also included.

The report is plain HTML without any images included, and therefore suitable for inclusion into e-mails or other documents that may be accessed outside the Hobbit system.

6.1.3 OPTIONS

-critical Report only on the statuses that are configured to show up on the **Critical Systems** view.

-old-nk-config Use the deprecated **NK** tag in `bb-hosts` to determine if tests appear on the Critical Systems view.

-env=FILENAME Loads the environment defined in FILENAME before executing the CGI script.

-area=NAME Load environment variables for a specific area. NB: if used, this option must appear before any `-env=FILENAME` option.

-debug Enables debugging output.

6.1.4 BUGS

Client-side configuration done in the `hobbit-clients.cfg(5)` is not currently reflected in the report.

Critical Systems view configuration is not reflected in the report.

6.1.5 SEE ALSO

`bb-hosts(5)`, `hobbit-alerts.cfg(5)`, `hobbit-clients.cfg(5)`, `hobbit(7)`

6.2 BB-REP.CGI

bb-rep.cgi - CGI front-end to bbgen reporting

6.2.1 SYNOPSIS

bb-rep.cgi [-noclean] [bbgen-options]

6.2.2 DESCRIPTION

bb-rep.cgi is invoked as a CGI script via the `bb-rep.sh` CGI wrapper. It triggers the generation of a Hobbit availability report for the timeperiod specified by the CGI parameters.

`bb-rep.cgi` is passed a `QUERY_STRING` environment variable with the following parameters:

`start-mon(Startmonthoftheresport) start-day(Startday-of-monthoftheresport) start-yr(Startyearoftheresport) end-mon(Endmonthoftheresport) end-day(Endday-of-monthoftheresport) end-yr(Endyearoftheresport) style(Reportstyle)` The following non-standard parameters are handled by the `bbgen` version of `bb-rep.cgi`:

`suburl(Pageinreporttogo,ifnotthetoppage)`

The “month” parameters must be specified as the three-letter english month name abbreviation: Jan, Feb, Mar ...

Start- and end-days are in the range 1..31; the start- and end-year must be specified including century (e.g. “2003”).

End-times beyond the current time are silently replaced with the current time.

The generated report will include data for the start- and end-days, i.e. the report will begin at 00:00:00 of the start-day, and end at 23:59:59 of the end-day.

The “style” parameter is passed directly to `bbgen(1)` and should be “crit”, “non-crit” or “all”. Other values result in undefined behaviour.

All of the processing involved in generating the report is done by invoking `bbgen(1)` with the proper “`-reportopts`” option.

6.2.3 OPTIONS

-noclean Do not clean the BBREP directory of old reports. Makes the report-tool go a bit faster - instead, you can clean up the BBREP directory e.g. via a cron-job.

-env=FILENAME Load the environment from FILENAME before executing the CGI.

bbgen-options All other options passed to `bb-rep.cgi` are passed on to the `bbgen(1)` program building the report files.

6.2.4 FILES

\$BBHOME/web/report_header HTML template header for the report request form

\$BBHOME/web/report_footer HTML template footer for the report request form

\$BBHOME/web/report_form HTML template report request form

6.2.5 ENVIRONMENT VARIABLES

BBGENREPOPTS `bbgen` options passed by default to the `bb-rep.cgi`. This happens in the `bb-rep.sh` wrapper.

BBHOME Home directory of the Hobbit server installation

BBREP Directory where generated reports are stored. This directory must be writable by the userid executing the CGI script, typically “www”, “apache” or “nobody”. Default: `$BBHOME/www/rep/`

BBREPURL The URL prefix to use when accessing the reports via a browser. Default: `$BBWEB/rep`

6.2.6 SEE ALSO

`bbgen(1)`, `bb-hosts(5)`, `hobbitserver.cfg(5)`

6.3 HOBBITD_FILESTORE

`hobbitd_filestore` - hobbitd worker module for storing Hobbit data

6.3.1 SYNOPSIS

```
hobbitd_channel --channel=status hobbitd_filestore --status [options]
hobbitd_channel --channel=data hobbitd_filestore --data [options]
hobbitd_channel --channel=notes hobbitd_filestore --notes [options]
hobbitd_channel --channel=enadis hobbitd_filestore --enadis [options]
```

6.3.2 DESCRIPTION

`hobbitd_filestore` is a worker module for `hobbitd`, and as such it is normally run via the `hobbitd_channel(8)` program. It receives `hobbitd` messages from a `hobbitd` channel via `stdin`, and stores these in the filesystem in a manner that is compatible with the Big Brother daemon, `bbd`.

This program can be started multiple times, if you want to store messages for more than one channel.

6.3.3 OPTIONS

- status** Incoming messages are “status” messages, they will be stored in the `$BBLOGS/` directory. If you are using `hobbit(7)` throughout your Hobbit server, you will not need to run this module to save status messages, unless you have a third-party add-on that reads the status-logs directly. This module is NOT needed to get trend graphs, you should run the `hobbitd_rrd(8)` module instead.
- data** Incoming messages are “data” messages, they will be stored in the `$BBDATA` directory. This module is not needed, unless you have a third-party module that processes the data-files. This module is NOT needed to get trend graphs, you should run the `hobbitd_rrd(8)` module instead.
- notes** Incoming messages are “notes” messages, they will be stored in the `$BBNOTES` directory. This modules is only needed if you want to allow people to remotely update the notes-files available on the Hobbit webpages.
- enadis** Incoming messages are enable/disable messages, they will update files in the `$BBDISABLED` directory. This is only needed if you have third-party add-ons that use these files.
- dir=DIRECTORY** Overrides the default output directory.
- html** Used together with “`-status`”. Tells `hobbitd_filestore` to also save an HTML version of the status-log. Should not be used unless you must run with “`BBLOGSTATUS=static`”.
- htmldir=DIRECTORY** The directory where HTML-versions of the status logs are stored. Default: `$BBHTML`
- htmlext=.EXT** Set the filename extension for generated HTML files. By default, HTML files are saved with a “`.html`” extension.
- multigraphs=TEST1[,TEST2]** This causes `hobbitd_filestore` to generate HTML status pages with links to service graphs that are split up into multiple images, with at most 5 graphs per image. If not specified, only the “disk” status is split up this way.
- only=test[,test,test]** Save status messages only for the listed set of tests. This can be useful if you have an external script that needs to parse some of the status logs, but you do not want to save all status logs.
- debug** Enable debugging output.

6.3.4 FILES

This module does not rely on any configuration files.

6.3.5 SEE ALSO

`hobbitd_channel(8)`, `hobbitd_rrd(8)`, `hobbitd(8)`, `hobbit(7)`

Chapter 7

Network Service Testing

7.1 BBTEST-NET

`bbtest-net` - Hobbit network test tool

7.1.1 SYNOPSIS

`bbtest-net -?`

`bbtest-net --help`

`bbtest-net --version`

`bbtest-net [options]`

(See the OPTIONS section for a description of the available commandline options).

7.1.2 DESCRIPTION

`bbtest-net(1)` handles the network tests of hosts defined in the Hobbit configuration file, `bb-hosts`. It is normally run at regular intervals by `hobbitlaunch(8)` via an entry in the `hobbitlaunch.cfg(5)` file.

`bbtest-net` does all of the normal tests of TCP-based network services (telnet, ftp, ssh, smtp, pop, imap) - i.e. all of the services listed as BBNETSVCS in `bbdef.sh`. For these tests, a completely new and very speedy service- checker has been implemented.

`bbtest-net` has built-in support for testing SSL-enabled protocols, e.g. imaps, pop3s, nntps, telnets, if SSL-support was enabled when configuring `bbgen`. The full list of known tests is found in the `bb-services(5)` file in `$BBHOME/etc/bb-services`.

In addition, it implements the “dns” and “dig” tests for testing DNS servers. This is done in the same way as `bb-network.sh` does it.

`bbtest-net` also implements a check for NTP servers - this test is called “ntp”. If you want to use it, you must define the `NTPDATE` environment variable to point at the location of your `ntpdate(1)` program.

Note: `bbtest-net` performs the connectivity test (ping) based on the hostname, unless the host is tagged with “testip” or the “–dns=ip” option is used. So the target of the connectivity test can be determined by your `/etc/hosts` file or DNS.

7.1.3 GENERAL OPTIONS

-timeout=N Determines the timeout (in seconds) for each service that is tested. For TCP tests (those from BBNETSVCS), if the connection to the service does not succeed within N seconds, the service is reported as being down. For HTTP tests, this is the absolute limit for the entire request to the webserver (the time needed to connect to the server, plus the time it takes the server to respond to the request). Default: 10 seconds

-conntimeout=N This option is deprecated, and will be ignored. Use the `-timeout` option instead.

-cmdtimeout=N This option sets a timeout for the external commands used for testing of NTP and RPC services, and to perform traceroute.

-concurrency=N Determines the number of network tests that run in parallel. Default is operating system dependent, but will usually be 256. If bbtest-net begins to complain about not being able to get a “socket”, try running bbtest-net with a lower value like 50 or 100.

-dns-timeout=N (default: 30 seconds) bbtest-net will timeout all DNS lookups after N seconds. Any pending DNS lookups are regarded as failed, i.e. the network tests that depend on this DNS lookup will report an error.

Note: If you use the `-no-ares` option, timeout of DNS lookups cannot be controlled by bbtest-net.

-dns-max-all=N Same as “`-dns-timeout=N`”. The “`-dns-max-all`” option is deprecated and should not be used.

-dns=[ip—only—standard] Determines how bbtest-net finds the IP addresses of the hosts to test. By default (the “standard”), bbtest-net does a DNS lookup of the hostname to determine the IP address, unless the host has the “testip” tag, or the DNS lookup fails.

With “`-dns=only`” bbtest-net will ONLY do the DNS lookup; if it fails, then all services on that host will be reported as being down.

With “`-dns=ip`” bbtest-net will never do a DNS lookup; it will use the IP address specified in `bb-hosts` for the tests. Thus, this setting is equivalent to having the “testip” tag on all hosts. Note that http tests will ignore this setting and still perform a DNS lookup for the hostname given in the URL; see the “`bbtest-net` tags for HTTP tests” section in `bb-hosts(5)`

-no-ares Disable the ARES resolver built into bbtest-net. This makes bbtest-net resolve hostnames using your system resolver function. You should only use this as a last resort if bbtest-net cannot resolve the hostnames you use in the normal way (via DNS or `/etc/hosts`). One reason for using this would be if you need to resolve hostnames via NIS/NIS+ (a.k.a. Yellow Pages).

The system resolver function does not provide a mechanism for controlling timeouts of the hostname lookups, so if your DNS or NIS server is down, bbtest-net can take a very long time to run. The `-dns-timeout` option is effectively disabled when using this option.

-dnslog=FILENAME Log failed hostname lookups to the file FILENAME. FILENAME should be a full pathname.

-report[=COLUMNNAME] With this option, bbtest-net will send a status message with details of how many hosts were processed, how many tests were generated, any errors that occurred during the run, and some timing statistics. The default columnname is “`bbtest`”.

-test-untagged When using the `BBLOCATION` environment variable to test only hosts on a particular network segment, bbtest-net will ignore hosts that do not have any “NET:x” tag. So only hosts that have a `NET:$BBLOCATION` tag will be tested.

With this option, hosts with no NET: tag are included in the test, so that all hosts that either have a matching NET: tag, or no NET: tag at all are tested.

-frequenttestlimit=N Used with the `bbretest-net.sh(1)` bbgen extension. This option determines how long failed tests remain in the frequent-test queue. The default is 1800 seconds (30 minutes).

-timelimit=N Causes bbtest-net to generate a warning if the run-time of bbtest-net exceeds N seconds. By default N is set to the value of `BBSLEEP`, so a warning triggers if the network tests cannot complete in the time given for one cycle of the BBNET server. Apart from the warning, this option has no effect, i.e. it will not terminate bbtest-net prematurely. So to eliminate any such warnings, use this option with a very high value of N.

-huge=N Warn if the response from a TCP test is more than N bytes. If you see from the bbtest status report that you are transferring large amounts of data for your tests, you can enable this option to see which tests have large replies. Default: 0 (disabled).

-validity=N Make the test results valid for N minutes before they go purple. By default test results are valid for 30 minutes; if you run bbtest-net less often than that, the results will go purple before the next run of bbtest-net. This option lets you change how long the status is valid.

7.1.4 OPTIONS FOR TESTS OF THE SIMPLE TCP SERVICES

-checkresponse[=COLOR] When testing well-known services (e.g. FTP, SSH, SMTP, POP-2, POP-3, IMAP, NNTP and rsync), bbtest-net will look for a valid service-specific “OK” response. If another response is seen, this will cause the test to report a warning (yellow) status. Without this option, the response from the service is ignored.

The optional color-name is used to select a color other than yellow for the status message when the response is wrong. E.g. “-checkresponse=red” will cause a “red” status message to be sent when the service does not respond as expected.

-no-flags By default, bbtest-net sends some extra information in the status messages, called “flags”. These are used by bbgen e.g. to pick different icons for reversed tests when generating the Hobbit webpages. This option makes bbtest-net omit these flags from the status messages.

7.1.5 OPTIONS FOR THE PING TEST

Note: bbtest-net uses the program defined by the FPING environment to execute ping-tests - by default, that is the *hobbitping(1)* utility. See *hobbitserver:cfg(5)* for a description of how to customize this, e.g. if you need to run it with “sudo” or a similar tool.

-ping Enables bbtest-net’s ping test. The column name used for ping test results is defined by the PINGCOLUMN environment variable in *hobbitserver:cfg(5)*.

If not specified, bbtest-net uses the CONNTEST environment variable to determine if it should perform the ping test or not. So if you prefer to use another tool to implement ping checks, either set the CONNTEST environment variable to false, or run bbtest-net with the “-noping”.

-noping Disable the connectivity test.

-trace

-notrace Enable/disable the use of traceroute when a ping-test fails. Performing a traceroute for failed ping tests is a slow operation, so the default is not to do any traceroute, unless it is requested on a per-host basis via the “trace” tag in the *bb-hosts(5)* entry for each host. The “-trace” option changes this, so the default becomes to run traceroute on all hosts where the ping test fails; you can then disable it on specific hosts by putting a “notrace” tag on the host-entry.

7.1.6 OPTIONS FOR HTTP (WEB) TESTS

-content=CONTENTTESTNAME Determines the name of the column Hobbit displays for content checks. The default is “content”. If you have used the “cont.sh” or “cont2.sh” scripts earlier, you may want to use “-content=cont” to report content checks using the same test name as these scripts do.

7.1.7 OPTIONS FOR SSL CERTIFICATE TESTS

-ssl=SSLCERTTESTNAME Determines the name of the column Hobbit displays for the SSL certificate checks. The default is “sslcert”.

-no-ssl Disables reporting of the SSL certificate check.

-sslwarn=N

-sslalarm=N Determines the number of days before an SSL certificate expires, where bbtest-net will generate a warning or alarm status for the SSL certificate column.

7.1.8 DEBUGGING OPTIONS

-no-update Don't send any status updates to the BBDISPLAY server. Instead, all messages are dumped to stdout.

-timing Causes bbtest-net to collect information about the time spent in different parts of the program. The information is printed on stdout just before the program ends. Note that this information is also included in the status report sent with the “-report” option.

-debug Dumps a bunch of status about the tests as they progress to stdout.

-dump[=before—=after—=both] Dumps internal memory structures before and/or after the tests have executed.

7.1.9 INFORMATIONAL OPTIONS

-help or -? Provide a summary of available commandline options.

-version Prints the version number of bbtest-net

-services Dump the list of defined TCP services bbtest-net knows how to test. Do not run any tests.

7.1.10 USING COOKIES IN WEB TESTS

If the file \$BBHOME/etc/cookies exist, cookies will be read from this file and sent along with the HTTP requests when checking websites. This file is in the Netscape Cookie format, see http://www.netscape.com/newsref/std/cookie_spec.html for details on this format. The *curl(1)* utility can output a file in this format if run with the “-cookie-jar FILENAME” option.

7.1.11 ABOUT SSL CERTIFICATE CHECKS

When bbtest-net tests services that use SSL- or TLS-based protocols, it will check that the server certificate has not expired. This check happens automatically for https (secure web), pop3s, imaps, nntps and all other SSL-enabled services (except ldap, see LDAP TESTS below).

All certificates found for a host are reported in one status message.

Note: On most systems, the end-date of the certificate is limited to Jan 19th, 2038. If your certificate is valid after this date, bbtest-net will report it as valid only until Jan 19, 2038. This is due to limitations in your operating system C library.

7.1.12 LDAP TESTS

ldap testing can be done in two ways. If you just put an “ldap” or “ldaps” tag in bb-hosts, a simple test is performed that just verifies that it is possible to establish a connection to the port running the ldap service (389 for ldap, 636 for ldaps).

Instead you can put an LDAP URI in bb-hosts. This will cause bbtest-net to initiate a full-blown LDAP session with the server, and do an LDAP search for the objects defined by the URI. This requires that bbtest-net was built with LDAP support, and relies on an existing LDAP library to be installed. It has been tested with OpenLDAP 2.0.26 (from Red Hat 9) and 2.1.22. The Solaris 8 system ldap library has also been confirmed to work for un-encrypted (plain ldap) access.

The format of LDAP URI's is defined in RFC 2255. LDAP URLs look like this:

```
\textbf{\{ldap://\}}
\emph{hostport}
\textbf{/}
\emph{dn}
[\textbf{?}]
\emph{attrs}
[\textbf{?}]
\emph{scope}
[\textbf{?}]
\emph{filter}
[\textbf{?}]
\emph{exts}
]]]
```

where:

```
\emph{hostport}
is a host name with an optional ":portnumber"
\emph{dn}
is the search base
\emph{attrs}
is a comma separated list of attributes to request
\emph{scope}
is one of these three strings:
base one sub (default=base)
\emph{filter}
is filter
\emph{exts}
are recognized set of LDAP and/or API extensions.
```

Example:

```
ldap://ldap.example.net/dc=example,dc=net?cn,sn?sub?(cn=*)
```

All “bind” operations to LDAP servers use simple authentication. Kerberos and SASL are not supported. If your LDAP server requires a username/password, use the “ldaplogin” tag to specify this, cf. *bb-hosts(5)*. If no user-name/password information is provided, an anonymous bind will be attempted.

SSL support requires both a client library and an LDAP server that support LDAPv3; it uses the LDAP “START-TLS” protocol request after establishing a connection to the standard (non-encrypted) LDAP port (usually port 389). It has only been tested with OpenSSL 2.x, and probably will not work with any other LDAP library.

The older LDAPv2 experimental method of tunnelling normal LDAP traffic through an SSL connection - ldaps, running on port 636 - is not supported, unless someone can explain how to get the OpenLDAP library to support it. This method was never formally described in an RFC, and implementations of it are non-standard.

For a discussion of the various ways of running encrypted ldap, see <http://www.openldap.org/lists/openldap-software/200305/msg00079.html> <http://www.openldap.org/lists/openldap-software/200305/msg00084.html> <http://www.openldap.org/lists/openldap-software/200201/msg00042.html> <http://www.openldap.org/lists/openldap-software/200206/msg00387.html>

When testing LDAP URI's, all of the communications are handled by the ldap library. Therefore, it is not possible to obtain the SSL certificate used by the LDAP server, and it will not show up in the “sslcert” column.

7.1.13 USING MULTIPLE NETWORK TEST SYSTEMS

If you have more than one system running network tests - e.g. if your network is separated by firewalls - then it is problematic to maintain multiple bb-hosts files for each of the systems. bbtest-net supports the NET:location tag in *bb-hosts(5)* to distinguish between hosts that should be tested from different network locations. If you set the environment variable BBLOCATION e.g. to "dmz" before running bbtest-net, then it will only test hosts that have a "NET:dmz" tag in bb-hosts. This allows you to keep all of your hosts in the same bb-hosts file, but test different sets of hosts by different BBNET systems.

7.1.14 BBTEST-NET INTERNALS

bbtest-net first reads the bb-services file to see which network tests are defined. It then scans the bb-hosts file, and collects information about the TCP service tests that need to be tested. It picks out only the tests that were listed in the bb-services file, plus the "dns", "dig" and "ntp" tests - those tests that bb-network.sh would normally use the "bbnet" tool to test.

It then runs two tasks in parallel: First, a separate process is started to run the "hobbitping" tool for the connectivity tests. While hobbitping is busy doing the "ping" checks, bbtest-net runs all of the TCP-based network tests.

All of the TCP-based service checks are handled by a connection tester written specifically for this purpose. It uses only standard Unix-style network programming, but relies on the Unix "select(2)" system-call to handle many simultaneous connections happening in parallel. Exactly how many parallel connections are being used depends on your operating system - the default is FD_SETSIZE/4, which amounts to 256 on many Unix systems.

You can choose the number of concurrent connections with the "--concurrency=N" option to bbtest-net.

Connection attempts timeout after 10 seconds - this can be changed with the "--timeout=N" option.

Both of these settings play a part in deciding how long the testing takes. A conservative estimate for doing N TCP tests is:

$$(1+(N/\text{concurrency}))*\text{timeout}$$

In real life it will probably be less, as the above formula is for every test to require a timeout. Since the most normal use of BB is to check for services that are active, you should have a lot less timeouts.

The "ntp" and "rpcinfo" checks rely on external programs to do each test. Thus, they perform only marginally better than the standard bb-network.sh script.

7.1.15 ENVIRONMENT VARIABLES

BBLOCATION Defines the network segment where bbtest-net is currently running. This is used to filter out only the entries in the *bb-hosts(5)* file that have a matching "NET:LOCATION" tag, and execute the tests for only those hosts.

BBMAXMSGSPERCOMBO Defines the maximum number of status messages that can be sent in one combo message. Default is 0 - no limit.

In practice, the maximum size of a single Hobbit message sets a limit - the default value for the maximum message size is 32 KB, but that will easily accomodate 100 status messages per transmission. So if you want to experiment with this setting, I suggest starting with a value of 10.

BBSLEEPBETWEENMSGS Defines a delay (in microseconds) after each message is transmitted to the BBDISPLAY server. The default is 0, i.e. send the messages as fast as possible. This gives your BBDISPLAY server some time to process the message before the next message comes in. Depending on the speed of your BBDISPLAY server, it may be necessary to set this value to half a second or even 1 or 2 seconds. Note that the value is specified in MICROseconds, so to define a delay of half a second, this must be set to the value "500000"; a delay of 1 second is achieved by setting this to "1000000" (one million).

FPING Command used to run the *hobbitping(1)* utility. Used by bbtest-net for connectivity (ping) testing. See *hobbitserver.cfg(5)* for more information about how to customize the program that is executed to do ping tests.

TRACEROUTE Location of the *traceroute(8)* utility, or an equivalent tool e.g. *mtr(8)*. Optionally used when a connectivity test fails to pinpoint the network location that is causing the failure.

NTPDATE Location of the *ntpdate(1)* utility. Used by bbtest-net when checking the “ntp” service.

RPCINFO Location of the *rpcinfo(8)* utility. Used by bbtest-net for the “rpc” service checks.

7.1.16 FILES

/server/etc/bb-services (Hobbit) This file contains definitions of TCP services that bbtest-net can test. Definitions for a default set of common services is built into bbtest-net, but these can be overridden or supplemented by defining services in the bb-services file. See *bb-services(5)* for details on this file.

\$BBHOME/etc/netrc - authentication data for password-protected webs If you have password-protected sites, you can put the usernames and passwords for these here. They will then get picked up automatically when running your network tests. This works for web-sites that use the “Basic” authentication scheme in HTTP. See *ftp(1)* for details - a sample entry would look like this

machinewww.acme.comloginfredpasswordWilma1 Note that the machine-name must be the name you use in the <http://machinename/> URL setting - it need not be the one you use for the system-name in Hobbit.

\$BBHOME/etc/cookies This file may contain website cookies, in the Netscape HTTP Cookie format. If a website requires a static cookie to be present in order for the check to complete, then you can add this cookie to this file, and it will be sent along with the HTTP request. To get the cookies into this file, you can use the “curl –cookie-jar FILE” to request the URL that sets the cookie.

\$BBTMP/*.status - test status summary Each time bbtest-net runs, if any tests fail (i.e. they result in a red status) then they will be listed in a file name TESTNAME.[LOCATION].status. The LOCATION part may be null. This file is used to determine how long the failure has lasted, which in turn decides if this test should be included in the tests done by *bbretest-net.sh(1)*

It is also used internally by bbtest-net when determining the color for tests that use the “badconn” or “badTEST-NAME” tags.

\$BBTMP/frequenttests.[LOCATION] This file contains the hostnames of those hosts that should be retested by the *bbretest-net.sh(1)* test tool. It is updated only by bbtest-net during the normal runs, and read by *bbretest-net.sh*.

7.1.17 SEE ALSO

bb-hosts(5), *bb-services(5)*, *hobbitserver.cfg(5)*, *hobbitping(1)*, *curl(1)*, *ftp(1)*, *fping(1)*, *ntpdate(1)*, *rpcinfo(8)*

7.2 BB-SERVICES

bb-services - Configuration of TCP network services

7.2.1 SYNOPSIS

\$BBHOME/etc/bb-services

7.2.2 DESCRIPTION

bb-services contains definitions of how *bbtest-net(1)* should test a TCP-based network service (i.e. all common network services except HTTP and DNS). For each service, a simple dialogue can be defined to check that the service is functioning normally, and optional flags determine if the service has e.g. a banner or requires SSL- or telnet-style handshaking to be tested.

7.2.3 FILE FORMAT

bb-services is a text file. A simple service definition for the SMTP service would be this:

```
[smtp]
send"mail\r\nquit\r\n"
expect"220"
optionsbanner
```

This defines a service called “smtp”. When the connection is first established, bbtest-net will send the string “mail\r\nquit\r\n” to the service. It will then expect a response beginning with “220”. Any data returned by the service (a so-called “banner”) will be recorded and included in the status message.

The full set of commands available for the bb-services file are:

[NAME] Define the name of the TCP service, which will also be the column-name in the resulting display on the test status. If multiple tests share a common definition (e.g. ssh, ssh1 and ssh2 are tested identically), you may list these in a single “[ssh—ssh1—ssh2]” definition, separating each service-name with a pipe-sign.

send STRING

expect STRING Defines the strings to send to the service after a connection is established, and the response that is expected. Either of these may be omitted, in which case *bbtest-net(1)* will simply not send any data, or match a response against anything.

The send- and expect-strings use standard escaping for non-printable characters. “\r” represents a carriage-return (ASCII 13), “\n” represents a line-feed (ASCII 10), “\t” represents a TAB (ASCII 8). Binary data is input as “\xNN” with NN being the hexadecimal value of the byte.

port NUMBER Define the default TCP port-number for this service. If no portnumber is defined, *bbtest-net(1)* will attempt to lookup the portnumber in the standard /etc/services file.

options option1[,option2 [,option3]] Defines test options. The possible options are

```
banner-includereceiveddatainthestatusmessage
ssl-serviceusesSSLsopermanSSLhandshake
telnet-serviceistelnet,soexchangetelnetoptions
```

7.2.4 FILES

\$BBHOME/etc/bb-services

7.2.5 SEE ALSO

`bbtest-net(1)`

7.3 HOBBITFETCH

7.3.1 NAME

hobbitfetch - fetch client data from passive clients

7.3.2 SYNOPSIS

hobbitfetch [-server=HOBBIT.SERVER.IP] [options]

7.3.3 DESCRIPTION

This utility is used to collect data from Hobbit clients.

Normally, Hobbit clients will themselves take care of sending all of their data directly to the Hobbit server. In that case, you do not need this utility at all. However, in some network setups clients may be prohibited from establishing a connection to an external server such as the Hobbit server, due to firewall policies. In such a setup you can configure the client to store all of the client data locally by enabling the *msgcache(8)* utility on the client, and using **hobbitfetch** on the Hobbit server to collect data from the clients.

hobbitfetch will only collect data from clients that have the **pulldata** tag listed in the *bb-hosts(5)* file. The IP-address listed in the bb-hosts file must be correct, since this is the IP-address where **hobbitfetch** will attempt to contact the client. If the msgcache daemon is running on a non-standard IP-address or portnumber, you can specify the portnumber as in **pulldata=192.168.1.2:8084** for contacting the msgcache daemon using IP 192.168.1.2 port 8084. If the IP-address is omitted, the default IP in the bb-hosts file is used. If the port number is omitted, the portnumber from the BBPORT setting in *hobbitserver.cfg(5)* is used (normally, this is port 1984).

7.3.4 OPTIONS

-server=HOBBIT.SERVER.IP Defines the IP address of the Hobbit server where the collected client messages are forwarded to. By default, messages are sent to the loopback address 127.0.0.1, i.e. to a Hobbit server running on the same host as **hobbitfetch**.

-interval=N Sets the interval (in seconds) between polls of a client. Default: 60 seconds.

-id=N Used when you have a setup with multiple Hobbit servers. In that case, you must run **hobbitfetch** on each of the Hobbit servers, with **hobbitfetch** instance using a different value of N. This allows several Hobbit servers to pick up data from the clients running msgcache, and msgcache can distinguish between which messages have already been forwarded to which server.

N is a number in the range 1-31.

-log-interval=N Limit how often **hobbitfetch** will log problems with fetching data from a host, in seconds. Default: 900 seconds (15 minutes). This is to prevent a host that is down or where msgcache has not been started from flooding the **hobbitfetch** logs. Note that this is ignored when debugging is enabled.

-debug Enable debugging output.

7.3.5 SEE ALSO

msgcache(8), *hobbitd(8)*, *hobbit(7)*

7.4 bbcmd

“/opt/bin/bbcmd” - Run a Hobbit command with environment set

7.4.1 SYNOPSIS

bbcmand –env=ENVFILE COMMAND

7.4.2 DESCRIPTION

bbcmand(1) is a utility that can setup the Hobbit environment variables as defined in a *hobbitlaunch(8)* compatible environment definition file, and then execute a command with this environment in place. It is mostly used for testing extension scripts or in other situations where you need to run a single command with the environment in place.

The “–env=ENVFILE” option points *bbcmand* to the file where the environment definitions are loaded from.

COMMAND is the command to execute after setting up the environment.

If you want to run multiple commands, it is often easiest to just use “sh” as the COMMAND - this gives you a sub-shell with the environment defined globally.

7.4.3 SEE ALSO

hobbitlaunch(8), *hobbit(7)*

7.5 CLIENTLAUNCH.CFG

clientlaunch.cfg - Task definitions for the hobbitlaunch utility

7.5.1 SYNOPSIS

hobbit/client/etc/clientlaunch.cfg

7.5.2 DESCRIPTION

The clientlaunch.cfg file holds the list of tasks that hobbitlaunch runs on a Hobbit client. This is typically just the Hobbit client itself, but you can add custom test scripts here and have them executed regularly by the Hobbit scheduler.

7.5.3 FILE FORMAT

See the *hobbitlaunch.cfg(5)* description.

7.5.4 SEE ALSO

hobbitlaunch(8), hobbit(7)

Chapter 8

Combination Tests

8.1 BBCOMBOTEST

`bbcombotest` - Hobbit combination test tool

8.1.1 SYNOPSIS

`bbcombotest --help bbcombotest --version bbcombotest [-debug] [-quiet]`

8.1.2 DESCRIPTION

`bbcombotest` is a Hobbit extension script that runs on the BBDISPLAY server. It combines the results of one or more of the normal Hobbit test results into a combined test result, using standard arithmetic or logical operators.

The resulting tests are sent to the Hobbit display server as any normal test - so all of the standard Hobbit functions (history, statistics etc.) are available for the combined tests.

The tool was born from the need to monitor systems with built-in redundancy and automatic failover - e.g. load-balanced web servers. But other uses are possible.

8.1.3 OPTIONS

-quiet Normally, the test status sent by `bbcombotest` includes information about the underlying test results used to determine the current value of the combined test. “`-quiet`” eliminates this information from the test status page.

-debug Provide debugging output for use in troubleshooting problems with `bbcombotest`.

-no-update Dont send any status messages - instead, the result of the combotests is simply dumped to stdout. Useful for debugging.

8.1.4 FILES

`“/etc/opt/hobbitserver42/bbcombotest.cfg”` Configuration file for `bbcombotest`, where the combined tests are defined

`“/etc/opt/hobbitserver42/hobbitlaunch.cfg”` Configuration file controlling when `bbcombotest` is run.

8.1.5 SEE ALSO

`bbcombotest.cfg(5)`, `bb-hosts(5)`, `hobbitserver.cfg(5)`, `hobbitlaunch.cfg(5)`

8.2 HOBBITGRAPH.CGI

8.2.1 NAME

`hobbitgraph.cgi` - CGI to generate Hobbit trend graphs

8.2.2 SYNOPSIS

`hobbitgraph [options]`

8.2.3 DESCRIPTION

`hobbitgraph.cgi` is invoked as a CGI script via the `hobbitgraph.sh` CGI wrapper.

`hobbitgraph.cgi` is passed a `QUERY_STRING` environment variable with the following parameters:

host Name of the host to generate a graph for

service Name of the service to generate a graph for

disp Display-name of the host, used on the generated graphs instead of hostname.

graph Can be “hourly”, “daily”, “weekly” or “monthly” to select the time period that the graph covers.

first Used to split multi-graphs into multiple graphs. This causes `hobbitgraph.cgi` to generate only the graphs starting with the “first’th” graph and continuing for “count”.

count Number of graphs in a multi-graph.

upper Set the upper limit of the graph. See `rrdgraph(1)` for a description of the “-u” option.

lower Set the lower limit of the graph. See `rrdgraph(1)` for a description of the “-l” option.

graph_start Set the starttime of the graph. This is used in zoom-mode.

graph_end Set the end-time of the graph. This is used in zoom-mode.

action=menu Generate an HTML page with links to 4 graphs, representing the hourly, weekly, monthly and yearly graphs. Doesn’t actually generate any graphs, only the HTML that links to the graphs.

action=selzoom Generate an HTML page with link to single graph, and with JavaScript code that lets the user select part of the graph for a zoom-operation. Doesn’t actually generate graph, only the HTML that links to the graph.

action=view Generate a single graph image.

8.2.4 OPTIONS

-config=FILENAME Loads the graph configuration file from FILENAME. If not specified, the file “`/etc/opt/hobbitserver42/hobbitgraph.cfg`” is used. See the `hobbitgraph.cfg(5)` for details about this file.

-env=FILENAME Loads the environment settings defined in FILENAME before executing the CGI.

-rrddir=DIRECTORY The top-level directory for the RRD files. If not specified, the directory given by the BBR-RDS environment is used.

-save=FILENAME Instead of returning the image via the CGI interface (i.e. on stdout), save the generated image to FILENAME.

-debug Enable debugging output.

8.2.5 ENVIRONMENT

QUERY_STRING Provided by the webserver CGI interface, this decides what graph to generate.

8.2.6 FILES

hobbitgraph.cfg: The configuration file determining how graphs are generated from RRD files.

8.2.7 SEE ALSO

`hobbitgraph.cfg(5)`, `hobbit(7)`, `rrdtool(1)`

8.3 BBCOMBOTEST.CFG

8.3.1 NAME

`bbcombotest.cfg` - Configuration of bbcombotest tool

8.3.2 SYNOPSIS

`"/etc/opt/hobbitserver42/bbcombotest.cfg"`

8.3.3 DESCRIPTION

`bbcombotest(1)` uses it's own configuration file, “`/etc/opt/hobbitserver42/bbcombotest.cfg`” Each line in this file defines a combined test.

8.3.4 FILE FORMAT

Each line of the file defines a new combined test. Blank lines and lines starting with a hash mark (#) are treated as comments and ignored.

The configuration file uses the hostnames and testnames that are already used in your Hobbit bb-hosts file. These are then combined using normal logical operators - “`—`” for “or”, “`&&`” for “and” etc.

A simple test - e.g. “`Web1.http`” - results in the value “1” if the “`http`” test for server “`Web1`” is green, yellow or clear. It yields the value “0” if it is red, purple or blue.

Apart from the logical operations, you can also do integer arithmetic and comparisons. E.g. the following is valid:
`WebCluster.http = (Web1.http + Web2.http + Web3.http) >= 2`

This test is green if two or more of the http tests for Web1, Web2 and Web3 are green.

The full range of operators are:

`+Add -Subtract *Multiply /Divide %Modulo —Bit-wise”or” &Bit-wise”and” —Logical”or” &&Logical”and”`
`>Greaterthan <Lessthan >=Greaterthanorequal <=Lessthanorequal ==Equal`

There is currently no support for a “not” operator. If you need it, use the transcription “`(host.test == 0)`” instead of “`!host.test`”.

NB: All operators have EQUAL PRECEDENCE. If you need something evaluated in a specific order, use parentheses to group the expressions together.

If the expression comes out as “0”, the combined test goes red. If it comes out as non-zero, the combined test is green.

Note: If the expression involves hostnames with a character that is also an operator - e.g. if you have a host “`t1-router-newyork.foo.com`” with a dash in the hostname - then the operator-character must be escaped with a backslash ‘\’ in the expression, or it will be interpreted as an operator. E.g. like this:

`nyc.conn=(t1\‐router\‐nyc.conn——backup\‐router\‐nyc.conn)`

8.3.5 EXAMPLE

```
WebCluster.http = (Web1.http — Web2.http) AppSrvCluster.procs = (AppSrv1.conn && AppSrv1.procs) —
(AppSrv2.conn && AppSrv2.procs) Customer.cluster = WebCluster.http && AppSrvCluster.procs
```

The first line defines a new test, with hostname “`WebCluster`” and the columnname “`http`”. It will be green if the `http` test on either the “`Web1`” or the “`Web2`” server is green.

The second line defines a “`procs`” test for the “`AppSrvCluster`” host. Each of the `AppSrv1` and `AppSrv2` hosts is checked for “`conn`” (ping) and their “`procs`” test. On each host, both of these must be green, but the combined test is green if that condition is fulfilled on just one of the hosts.

The third line uses the two first tests to build a “double combined” test, defining a test that shows the overall health of the system.

8.3.6 FILES

`“/etc/opt/hobbitserver42/bbcombotest.cfg”`

8.3.7 SEE ALSO

`bbcombotest(1)`

8.4 HOBBITGRAPH.CGI

8.4.1 NAME

`hobbitgraph.cgi` - CGI to generate Hobbit trend graphs

8.4.2 SYNOPSIS

`hobbitgraph [options]`

8.4.3 DESCRIPTION

`hobbitgraph.cgi` is invoked as a CGI script via the `hobbitgraph.sh` CGI wrapper.

`hobbitgraph.cgi` is passed a `QUERY_STRING` environment variable with the following parameters:

host Name of the host to generate a graph for

service Name of the service to generate a graph for

disp Display-name of the host, used on the generated graphs instead of hostname.

graph Can be “hourly”, “daily”, “weekly” or “monthly” to select the time period that the graph covers.

first Used to split multi-graphs into multiple graphs. This causes `hobbitgraph.cgi` to generate only the graphs starting with the “first’th” graph and continuing for “count”.

count Number of graphs in a multi-graph.

upper Set the upper limit of the graph. See `rrdgraph(1)` for a description of the “-u” option.

lower Set the lower limit of the graph. See `rrdgraph(1)` for a description of the “-l” option.

graph_start Set the starttime of the graph. This is used in zoom-mode.

graph_end Set the end-time of the graph. This is used in zoom-mode.

action=menu Generate an HTML page with links to 4 graphs, representing the hourly, weekly, monthly and yearly graphs. Doesn’t actually generate any graphs, only the HTML that links to the graphs.

action=selzoom Generate an HTML page with link to single graph, and with JavaScript code that lets the user select part of the graph for a zoom-operation. Doesn’t actually generate graph, only the HTML that links to the graph.

action=view Generate a single graph image.

8.4.4 OPTIONS

-config=FILENAME Loads the graph configuration file from FILENAME. If not specified, the file “`/etc/opt/hobbitserver42/hobbitgraph.cfg`” is used. See the `hobbitgraph.cfg(5)` for details about this file.

-env=FILENAME Loads the environment settings defined in FILENAME before executing the CGI.

-rrddir=DIRECTORY The top-level directory for the RRD files. If not specified, the directory given by the BBR-RDS environment is used.

-save=FILENAME Instead of returning the image via the CGI interface (i.e. on stdout), save the generated image to FILENAME.

-debug Enable debugging output.

8.4.5 ENVIRONMENT

QUERY_STRING Provided by the webserver CGI interface, this decides what graph to generate.

8.4.6 FILES

hobbitgraph.cfg: The configuration file determining how graphs are generated from RRD files.

8.4.7 SEE ALSO

`hobbitgraph.cfg(5)`, `hobbit(7)`, `rrdtool(1)`

8.5 BB-SNAPSHOT.CGI

8.5.1 NAME

bb-snapshot.cgi - CGI program to rebuild the Hobbit webpages for a specific point in time.

8.5.2 SYNOPSIS

bb-snapshot.cgi

8.5.3 DESCRIPTION

bb-snapshot.cgi is invoked as a CGI script via the **bb-snapshot.sh** CGI wrapper. It rebuilds the Hobbit web pages to the look they had at a particular point in time, based upon the historical information logged about events.

bb-snapshot.cgi is passed a **QUERY_STRING** environment variable with the following parameters:

mon(Startmonthofthesnapshot) day(Startday-of-monthofthesnapshot) yr(Startyearofthesnapshot) hour(Starthourofthesnapshot)
min(Startminuteofthesnapshot) sec(Startsecondofthesnapshot)

The “month” parameters must be specified as the three-letter english month name abbreviation: Jan, Feb, Mar ...

“day” must be in the range 1..31; “yr” must be specified including century (e.g. “2003”). “hour” must be specified using a 24-hour clock.

All of the processing involved in generating the report is done by invoking **bbgen(1)** with the proper “–snapshot” option.

8.5.4 OPTIONS

-env=FILENAME Load environment from FILENAME before executing the CGI.

bbgen-options All options except “–env” are passed on to the **bbgen(1)** program building the snapshot files.

8.5.5 ENVIRONMENT VARIABLES

BBGENSNAPOPTS bbgen options passed by default to the **bb-snapshot.cgi** script. This happens in the **bb-snapshot.sh** CGI wrapper script.

BBHOME Home directory of the Hobbit server files

BBSNAP Directory where generated snapshots are stored. This directory must be writable by the userid executing the CGI script, typically “www”, “apache” or “nobody”. Default: \$BBHOME/www/snap/

BBSNAPURL The URL prefix to use when accessing the reports via a browser. Default: \$BBWEB/snap

8.5.6 SEE ALSO

bbgen(1), **bb-hosts(5)**, **hobbitserver.cfg(5)**

8.6 HOBBITD_SAMPLE

8.6.1 NAME

hobbitd_sample - example of a hobbitd worker module

8.6.2 SYNOPSIS

hobbitd_channel –channel=status hobbitd_sample [options]

8.6.3 DESCRIPTION

hobbitd_sample is a worker module for hobbitd, and as such it is normally run via the *hobbitd_channel(8)* program. It receives messages from hobbitd via stdin, and simply displays these on stdout. It can be used with all types of hobbitd channels.

hobbitd_sample is not designed to actually run, except as a demonstration. The purpose of this tool is to show how hobbitd worker modules can be implemented to handle different tasks that need to hook into the hobbitd processing.

8.6.4 OPTIONS

-timeout=N Read messages with a timeout of N seconds.

-debug Enable debugging output.

8.6.5 SEE ALSO

hobbitd_channel(8), hobbitd(8), hobbit(7)

8.7 BB-CSVINFO.CGI

8.7.1 NAME

`bb-csvinfo.cgi` - CGI program to show host information from a CSV file

8.7.2 SYNOPSIS

`bb-csvinfo.cgi`

8.7.3 DESCRIPTION

`bb-csvinfo.cgi` is invoked as a CGI script via the `bb-csvinfo.sh` CGI wrapper. Based on the parameters it receives, it searches a comma-separated file for the matching host, and presents the information found as a table.

`bb-csvinfo.cgi` is passed a `QUERY_STRING` environment variable with the following parameters:

`key(stringtosearchfor,typicallyhostname) column(columnnumbertosearch-default0) db(nameoftheCSVdatabasefilein$BBHOME/etc/delimiter(delimitercharacterforcolumns,defaultsemi-colon)`

CSV files are easily created from e.g. spreadsheets, by exporting them in CSV format. You should have one host per line, with the first line containing the column headings. Despite their name, the default delimiter for CSV files is the semi-colon - if you need a different delimiter, invoke `bb-csvinfo.cgi` with the “`delimiter=<character>`” in the query string.

8.7.4 Example usage

This example shows how you can use the `bb-csvinfo` CGI. It assumes you have a CSV-formatted file with information about the hosts stored as “`/etc/opt/hobbitserver42/hostinfo.csv`”, and the hostname is in the first column of the file.

Use with the `bbgen -docurl` The `-docurl` option to `bbgen(1)` sets up all of the hostnames on your Hobbit webpages to act as links to a CGI script. To invoke the `bb-csvinfo` CGI script, run `bbgen` with the option

`-docurl=/cgi-bin/bb-csvinfo.sh?db=hostinfo.csv&key=%s`

8.7.5 SEE ALSO

`bb-hosts(5)`, `hobbitserver.cfg(5)`, `bbgen(1)`

8.8 HOBBIT-HOSTGRAPHS.CGI

hobbit-hostgraphs.cgi - CGI program to show multiple graphs

8.8.1 SYNOPSIS

hobbit-hostgraph.cgi

8.8.2 DESCRIPTION

hobbit-hostgraph.cgi is invoked as a CGI script via the `hobbit-hostgraph.sh` CGI wrapper.

If no parameters are provided when invoked, it will present a form where the user can select a time period, one or more hosts, and a set of graphs.

The parameters selected by the user are passed to a second invocation of `hobbit-hostgraph.cgi`, and result in a webpage showing a list of graph images based on the trend data stored about the hosts.

If multiple graph-types are selected, `hobbit-hostgraph.cgi` will display a list of graphs, with one graph per type.

If multiple hosts are selected, `hobbit-hostgraph.cgi` will attempt to display a multi-host graph for each type where the graphs for all hosts are overlayed in a single image, allowing for easy comparison of the hosts.

The hostlist uses the `PAGEPATH` cookie provided by Hobbit webpages to select the list of hosts to present. Only the hosts visible on the page where `hobbit-hostgraph.cgi` is invoked from will be visible.

The resulting graph page can be bookmarked, but the bookmark also fixates the time period shown.

8.8.3 OPTIONS

-env=FILENAME Loads the environment defined in FILENAME before executing the CGI script.

8.8.4 BUGS

This utility is experimental. It may change in a future release of Hobbit.

It is possible for the user to select graphs which do not exist. This results in broken image links.

The set of graph-types is fixed in the `server/web/hostgraphs_form` template and does not adjust to which graphs are available.

If the tool is invoked directly, all hosts defined in Hobbit will be listed.

8.8.5 SEE ALSO

`bb-hosts(5)`, `hobbitserver.cfg(5)`

8.9 BB-DATEPAGE.CGI

`bb-datepage.cgi` - Hobbit CGI script to view pre-built reports by date

8.9.1 SYNOPSIS

`bb-datepage.cgi?type=day,week,month -url=URLPREFIX [options]`

8.9.2 DESCRIPTION

bb-datepage.cgi is invoked as a CGI script via the `bb-datepage.sh` CGI wrapper.

`bb-datepage.cgi` is passed a `QUERY_STRING` environment variable with the type of time-selection that is desired: Either “day”, “week” or “month” can be requested. It will then generate a web form with appropriate day/week/month selection boxes, and based on the users’ selection a resulting url is built from the `URLPREFIX` and the time selection. The browser is then redirected to this URL.

The URL is constructed from the `URLPREFIX`, the type-parameter, the value of the “pagepath” or “host” cookie, and the users’ selection as follows:

type=day The final URL is `URLPREFIX/daily/YEAR/MONTH/DAY/PAGEPATH`.

type=week The final URL is `URLPREFIX/weekly/YEAR/WEEK/PAGEPATH`.

type=month The final URL is `URLPREFIX/monthly/YEAR/MONTH/PAGEPATH`.

`YEAR` is the full year (4 digits, including century). `MONTH` is the two-digit number of the month (01..12).

`DAY` is the number of the day in the month (01..31). `WEEK` is the ISO 8601:1988 week-number (01..53).

`PAGEPATH` is the current value of the “pagepath” cookie if set; if it is not set but the “host” cookie is set, then this host is looked up in the `bb-hosts` file and the page where this host is found is used for `PAGEPATH`. These two cookies are set by the default web-header templates supplied with Hobbit.

8.9.3 OPTIONS

-url=URLPREFIX This specifies the initial part of the final URL. This option is required.

-hfile=FILENAME Specifies the template files (from `$BBHOME/web/`) to use. The default is “`-hfile=report`”.

-color=COLOR Sets the background color of the generated webpage. The default is blue.

-env=FILENAME Loads the environment defined in `FILENAME` before executing the CGI script.

-debug Enables debugging output.

\$BBHOME/web/report_form.daily HTML form template for the date selection form when `type=daily`.

\$BBHOME/web/report_form.weekly HTML form template for the date selection form when `type=weekly`.

\$BBHOME/web/report_form.monthly HTML form template for the date selection form when `type=monthly`.

\$BBHOME/web/report_header HTML header file for the generated web page

\$BBHOME/web/report_footer HTML footer file for the generated web page

8.9.4 ENVIRONMENT VARIABLES

BBHOME Used to locate the template files for the generated web pages.

QUERY_STRING Contains the parameters for the CGI script.

8.9.5 SEE ALSO

`bbgen(1)`, `bb-hosts(5)`, `hobbitserver.cfg(5)`

8.10 BBDIGEST

`bbdigest` - calculate message digests

8.10.1 SYNOPSIS

`bbdigest md5—sha1—rmd160 [filename]`

8.10.2 DESCRIPTION

bbdigest(1) is a utility to calculate message digests for a file or document. It is used when defining HTTP- or FTP-based content checks, where *bbtest-net(1)* checks that a URL returns a specific document; instead of having to compare the entire document, the comparison is done against a pre-computed message digest value using the MD5, RIPEMD160, SHA1 or any of the SHA2 (SHA-512, SHA-256, SHA-384, SHA-224) message digest algorithms.

The optional **filename** parameter is the input file whose message digest should be calculated; if no filename is given, the data is read from standard input.

`bbdigest` outputs a string containing the digest algorithm and the computed message digest. This is in a format suitable for use in the *bb-hosts(5)* definition of a content check.

8.10.3 EXAMPLE

```
$ bbdigestmd5index.html md5:88b81b110a85c83db56a939caa2e2cf6
$ curl-shttp://www.foo.com/—bbdigestsha1 sha1:e5c69784cb971680e2c7380138e04021a20a45a2
```

8.10.4 SEE ALSO

`bbtest-net(1)`, `bb-hosts(5)`

8.11 HOBBITLAUNCH.CFG

hobbitlaunch.cfg - Task definitions for the hobbitlaunch utility

8.11.1 SYNOPSIS

`"/etc/opt/hobbitserver42/hobbitlaunch.cfg"`

8.11.2 DESCRIPTION

The hobbitlaunch.cfg file holds the list of tasks that hobbitlaunch runs to perform all of the tasks needed by the Hobbit monitor.

8.11.3 FILE FORMAT

A task is defined by a **key** , a **command** , and optionally also **interval** , **environment** , and **logfile** .

Blank lines and lines starting with a hash mark (#) are treated as comments and ignored. Long lines can be broken up by putting a backslash at the end of the line and continuing the entry on the next line.

An entry looks like this:

```
[hobbitd]
ENVFILE/usr/local/hobbit/server/etc/hobbitserver.cfg
CMD/usr/local/hobbit/server/bin/hobbitd
[updateweb]
ENVFILE/usr/local/hobbit/server/etc/hobbitserver.cfg
CMD/usr/local/hobbit/server/bin/bbgen
NEEDShobbitd
GROUPwebupdates
INTERVAL5m
LOGFILE/var/log/hobbit/updateweb.log
```

The **key** is enclosed in angle brackets, and must be unique for each task. You can choose your key-names as you like, they are only used internally in hobbitlaunch to identify each task.

The **command** is defined by the CMD keyword. This is the full command including any options you want to use for this task. This is required for all tasks.

The **DISABLED** keyword means that this command is disabled. hobbitlaunch will not start this task. It is recommended that you use this to disable standard tasks, instead of removing them or commenting them out. Upgrades to Hobbit will add standard tasks back into the file, so unless you have them listed as DISABLED then tasks may re-appear unexpectedly after an upgrade.

The **ONHOST** keyword tells hobbitlaunch that this task should only run on specific hosts. After the ONHOST keyword, you must provide a “regular expression”; if the hostname where hobbitlaunch runs matches this expression, then the task will run. If it doesn’t match, then the task is treated as if it were DISABLED.

The **NEEDS** instructs hobbitlaunch not to run this task unless the task defined by the NEEDS keyword is already running. This is used e.g. to delay the start of some application until the needed daemons have been started. The task that must be running is defined by its **key**

The **GROUP** keyword can be used to limit the number of tasks that may run simultaneously. E.g. if you are generating multiple pagesets of webpages, you don't want them to run at the same time. Putting them into a GROUP will cause hobbitlaunch to delay the start of new tasks, so that only one task will run per group. You can change the limit by defining the group before the tasks, with a “GROUP groupname maxtasks” line.

The **INTERVAL** keyword defines how often this command is executed. The example shows a command that runs every 5 minutes. If no interval is given, the task is only run once - this is useful for tasks that run continually as daemons - although if the task stops for some reason, then hobbitlaunch will attempt to restart it. Intervals can be specified in seconds (if you just put a number there), or in minutes (5m), hours (2h), or days (1d).

The **ENVFILE** setting points to a file with definitions of environment variables. Before running the task, hobbit-launch will setup all of the environment variables listed in this file. Since this is a per-task setting, you can use the same hobbitlaunch instance to run e.g. both the server- and client-side Hobbit tasks. If this option is not present, then the environment defined to hobbitlaunch is used.

The **ENVAREA** setting modifies which environment variables are loaded, by picking up the ones that are defined for this specific “area”. See *hobbitserver.cfg(5)* for information about environment areas.

The **LOGFILE** setting defines a logfile for the task. hobbitlaunch will start the task with stdout and stderr redirected to this file. If this option is not present, then the output goes to the same location as the hobbitlaunch output.

8.11.4 SEE ALSO

`hobbitlaunch(8)`, `hobbitd(8)`, `hobbit(7)`

8.12 BB-EVENTLOG.CGI

bb-eventlog.cgi - CGI program to report the Hobbit eventlog

8.12.1 SYNOPSIS

bb-eventlog.cgi

8.12.2 DESCRIPTION

bb-eventlog.cgi is invoked as a CGI script via the bb-eventlog.sh CGI wrapper. Based on the parameters it receives, it generates the Hobbit event log for a period. This log shows all status changes that have occurred for all hosts and services.

bb-eventlog.cgi is passed a QUERY_STRING environment variable with the following parameters:
MAXTIME(maximumminutestogobackinthelog) MAXCOUNT(maximumnumberofeventstoreport)

8.12.3 OPTIONS

-top=N Shows a summary of the top-N changing hosts and services.

-env=FILENAME Loads the environment defined in FILENAME before executing the CGI script.

8.12.4 SEE ALSO

bb-hosts(5), hobbitserver.cfg(5)

8.13 HOBBIT-MAILACK

hobbit-mailack - permit acknowledging alerts via e-mail

8.13.1 SYNOPSIS

hobbit-mailack **-env=FILENAME** [**-debug**]

8.13.2 DESCRIPTION

hobbit-mailack normally runs as an input mail-filter for the hobbit user, e.g. by being called from the hobbit users' *procmailrc(5)* file. hobbit-mailack recognizes e-mails that are replies to *hobbitd_alert(8)* mail alerts, and converts the reply mail into an acknowledge message that is sent to the Hobbit system. This permits an administrator to acknowledge an alert via e-mail.

8.13.3 ADDING INFORMATION TO THE REPLY MAIL

By default, an acknowledgment is valid for 1 hour. If you know in advance that solving the problem is going to take longer, you can change this by adding **delay=DURATION** to the subject of your mail reply or on a line in the reply message. Duration is in minutes, unless you add a trailing 'h' (for 'hours'), 'd' (for 'days') or 'w' (for 'weeks').

You can also include a message that will show up on the status-page together with the acknowledgment, e.g. to provide an explanation for the issue or some other information to the users. You can either put it at the end of the subject line as **msg=Some random text**, or you can just enter it in the e-mail as the first non-blank line of text in the mail (a "delay=N" line is ignored when looking for the message text).

8.13.4 USE WITH PROCMAIL

To setup hobbit-mailack, create a **.procmailrc** file in the hobbit-users home-directory with the following contents:

```
DEFAULT=$HOME/Mailbox
LOGFILE=$HOME/procmail.log
:0
| $HOME/server/bin/hobbit-mailack --env=$HOME/server/etc/hobbitserver.cfg
```

8.13.5 USE WITH QMAIL

If you are using Qmail to deliver mail locally, you can run hobbit-mailack directly from a **.qmail** file. Setup the hobbit-users **.qmail** file like this:

```
| $HOME/server/bin/hobbit-mailack --env=$HOME/server/etc/hobbitserver.cfg
```

8.13.6 OPTIONS

-env=FILENAME Load environment from FILENAME, usually *hobbitserver.cfg*.

-debug Dont send a message to *hobbitd*, but dump the message to stdout.

8.13.7 SEE ALSO

hobbitd_alert(8), *hobbitd(8)*, *hobbit(7)*

8.14 BB-FINDHOST.CGI

bb-findhost.cgi - Hobbit CGI script to find hosts

8.14.1 SYNOPSIS

bb-findhost.cgi?host=REGEX

8.14.2 DESCRIPTION

bb-findhost.cgi is invoked as a CGI script via the bb-findhost.sh CGI wrapper.

bb-findhost.cgi is passed a QUERY_STRING environment variable with the “host=REGEX” parameter. The REGEX is a Posix regular expression (see *regex(7)*) describing the hostnames to look for. A trailing wildcard is assumed on all hostnames - e.g. requesting the hostname “www” will match any host whose name begins with “www”.

It then produces a single web page, listing all of the hosts that matched any of the hostnames, with links to the Hobbit webpages where they are located.

The output page lists hosts in the order they appear in the *bb-hosts(5)* file.

A sample web page implementing the search facility is included with bbgen, you access it via the URL /bb/help/bb-findhost.html.

8.14.3 OPTIONS

-env=FILENAME Loads the environment from FILENAME before executing the CGI.

8.14.4 FILES

\$BBHOME/web/findhost_header HTML header file for the generated web page

\$BBHOME/web/findhost_footer HTML footer file for the generated web page

\$BBHOME/web/findhost_form Query form displayed when bb-findhost.cgi is called with no parameters.

8.14.5 ENVIRONMENT VARIABLES

BBHOSTS bb-findhost.cgi uses the BBHOSTS environment variable to find the bb-hosts file listing all known hosts and their page locations.

BBHOME Used to locate the template files for the generated web pages.

8.14.6 SEE ALSO

bbgen(1), bb-hosts(5), hobbitserver.cfg(5)

8.15 HOBBITCLIENT.CFG

hobbitclient.cfg - Hobbit client environment variables

8.15.1 DESCRIPTION

Hobbit programs use multiple environment variables beside the normal set of variables. For the Hobbit client, the environment definitions are stored in the `hobbit/client/etc/hobbitclient.cfg` file. Each line in this file is of the form **NAME=VALUE** and defines one environment variable NAME with the value VALUE.

8.15.2 SETTINGS

BBDISP The IP-address used to contact the Hobbit server. Default: Chosen when the Hobbit client was compiled.

BBDISPLAYS List of IP-adresses of Hobbit servers. Data will be sent to all of the servers listed here. This setting is only used if `BBDISP=0.0.0.0`.

BBPORT The portnumber for used to contact the Hobbit server. Default: 1984.

BBHOME The Hobbit client top-level directory. Default: The `$HOBBITCLIENTHOME` setting inherited from the “runclient.sh” script which starts the Hobbit client.

BBCLIENTLOGS The directory for the Hobbit clients’ own logfiles. Default: `$BBHOME/logs`

BBTMP Directory used for temporary files. Default: `$BBHOME/tmp/`

BB Full path to the `bb(1)` client program. Default: `$BBHOME/bin/bb`.

Commands Many extension scripts expect a series of environment variables to point at various system utilities. These are included in the file when the client is built.

8.15.3 INHERITED SETTINGS

Some environment variables are inherited from the “runclient.sh” script which launches the Hobbit client:

MACHINEDOTS The hostname of the local system. Default: Taken from “`uname -n`”.

MACHINE The hostname of the local system, with dots replaced by commas. For compatibility with Big Brother extension scripts.

BBOSTYPE The operating system of the local system, in lowercase. Default: taken from “`uname -s`”.

HOBBITCLIENTHOME The top-level directory for the Hobbit client. Default: The location of the “runclient.sh” script.

8.15.4 SEE ALSO

`hobbit(7)`

Chapter 9

Hobbit/BB Messages Protocols

9.1 BNF

Extended Backus.Naur Form (http://en.wikipedia.org/wiki/Backus-Naur_form BNF) is a metasyntax used to express context-free grammars: that is, a formal way to describe formal languages. We use BNF to describe the formal language in used between hobbit client and server.

9.1.1 Hobbit messages syntax

```
<hb-messages> ::= <status_msg> <notify_msg> <data_msg> <enable_msg> <query_msg>
                 <config_msg> <drop_msg> <rename_msg> <hobbidlog_msg> <hobbitdxlog_msg>
                 <hobbitboard_msg> <download_msg> <client_msg> <clientlog_msg> <ping_msg>
                 <pullclient_msg> <ghostlist_msg> <schedule_msg>

<drop_msg>   ::= <drop HOSTNAME> <EOL> | <drop HOSTNAME TESTNAME> <EOL>
<HOSTNAME>   ::= <a..z>
<TESTNAME>   ::= <cpu|trends|conn|info|>
<status_msg> ::= <status [+LIFETIME] [/group:GROUP] HOSTNAME.TESTNAME COLOR |
                  <additional text> ] > <EOL>
<notify_msg> ::= <notify HOSTNAME.TESTNAME <message text> > >
<data_msg>    ::= <data HOSTNAME.DATANAME <newline> <additional text> >
<enable_msg>  ::= [enable HOSTNAME.TESTNAME]
<query_msg>   ::= [query HOSTNAME.TESTNAME]
<config_msg>  ::= [config FILENAME]
<drop_msg>    ::= [drop HOSTNAME]
<rename_msg>  ::= [rename OLDHOSTNAME NEWHOSTNAME]
<hobbidlog_msg> ::= 
<hobbitdxlog_msg> ::= [hobbitdxlog HOSTNAME.TESTNAME]
<hobbitboard_msg> ::= [hobbitboard [CRITERIA] [fields=FIELDLIST]]
<download_msg> ::= [download FILENAME] <EOL>
<client_msg>   ::= [client HOSTNAME.OSTYPE [HOSTCLASS]]
<clientlog_msg> ::= [clientlog HOSTNAME [section=SECTIONNAME[, SECTIONNAME...]]]
<ping_msg>    ::= [ping] <EOL>
<pullclient_msg> ::= [pullclient] <EOL>
<ghostlist_msg> ::= [ghostlist] <EOL>
<schedule_msg> ::= [schedule [TIMESTAMP COMMAND]] <EOL>
<SECTIONNAME> ::= [cpu|la| others sectionname]
```

9.1.2 Big Brother messages syntax

```

<bb-messages> ::= <status_msg> <notify_msg> <data_msg> <enable_msg> <query_msg>
                 <config_msg> <drop_msg> <rename_msg> <hobbitlog_msg> <hobbitxlog_msg>
                 <hobbitboard_msg> <download_msg> <client_msg> <clientlog_msg> <ping_msg>
                 <pullclient_msg> <ghostlist_msg> <schedule_msg>

<drop_msg>   ::= <drop HOSTNAME> <EOL> | <drop HOSTNAME TESTNAME> <EOL>
<HOSTNAME>   ::= <a...z>
<TESTNAME>   ::= <cpu|trends|conn|info|>
<status_msg> ::= <status[+LIFETIME] [/group:GROUP] HOSTNAME.TESTNAME COLOR |
                  <additional text> ]> <EOL>
<notify_msg> ::= <notify HOSTNAME.TESTNAME <message text> ] >
<data_msg>   ::= <data HOSTNAME.DATANAME <newline> <additional text> >
<enable_msg> ::= [enable HOSTNAME.TESTNAME]
<query_msg>  ::= [query HOSTNAME.TESTNAME]
<config_msg> ::= [config FILENAME]
<drop_msg>   ::= [drop HOSTNAME]
<rename_msg> ::= [rename OLDHOSTNAME NEWHOSTNAME]
<hobbitlog_msg> ::= 
<hobbitxlog_msg> ::= [hobbitxlog HOSTNAME.TESTNAME]
<hobbitboard_msg> ::= [hobbitboard [CRITERIA] [fields=FIELDLIST]]
<download_msg> ::= [download FILENAME] <EOL>
<client_msg>  ::= [client HOSTNAME.OSTYPE [HOSTCLASS]]
<clientlog_msg> ::= [clientlog HOSTNAME [section=SECTIONNAME[, SECTIONNAME...]]]
<ping_msg>   ::= [ping] <EOL>
<pullclient_msg> ::= [pullclient] <EOL>
<ghostlist_msg> ::= [ghostlist] <EOL>
<schedule_msg> ::= [schedule [TIMESTAMP COMMAND]] <EOL>
<SECTIONNAME> ::= [cpu|la| others sectionname]
```

9.2 bb - Hobbit client communication program

9.2.1 SYNOPSIS

`“/opt/bin/bb” [options] RECIPIENT message`

9.2.2 DESCRIPTION

bb(1) is the client program used to communicate with a Hobbit server. It is frequently used by Hobbit client systems to send in status messages and pager alerts on local tests.

In Hobbit, the *bb* program is also used for administrative purposes, e.g. to rename or delete hosts, or to disable hosts that are down for longer periods of time.

9.2.3 OPTIONS AND PARAMETERS

-debug Enable debugging. This prints out details about how the connection to the BBDISPLAY server is being established.

-proxy=http://PROXYSERVER:PROXPORT/ When sending the status messages via HTTP, use this server as an HTTP proxy instead of connecting directly to the BBDISPLAY server.

-timeout=N Specifies the timeout for connecting to the Hobbit server, in seconds. The default is 5 seconds.

RECIPIENT The **RECIPIENT** parameter defines which server receives the message. If **RECIPIENT** is given as “0.0.0.0”, then the message is sent to all of the servers listed in the BBDISPLAYS or BBPAGERS environment variable (for “status” and “page” messages, respectively).

Usually, a client will use “\$BBDISP” for the **RECIPIENT** parameter, as this is defined for the client scripts to automatically contain the correct value.

The **RECIPIENT** parameter may be a URL for a webserver that has the *bbmessage.cgi* or similar script installed. This tunnels the Hobbit messages to the BBDISPLAY server using standard HTTP protocol. The *bbmessage.cgi(8)* CGI tool (included in Hobbit) must be installed on the webserver for the HTTP transport to work.

MESSAGE The **message** parameter is the message to be sent across to the Hobbit server. Messages must be enclosed in quotes, but by doing so they can span multiple lines. The maximum size of a message is defined by the maximum allowed length of your shell’s commandline, and is typically 8-32 KB.

If you need to send longer status messages, you can specify “@” as the message: *bb* will then read the status message from its *stdin*.

9.2.4 HOBBIT MESSAGE SYNTAX

This section lists the most commonly used messages in the Hobbit protocol.

status[+LIFETIME] [/group:GROUP] HOSTNAME.TESTNAME COLOR <additional text> This sends in a status message for a single test (column) on a single host. TESTNAME is the name of the column where this test will show up; any name is valid except that using dots in the testname will not work. COLOR must be one of the valid colors: “green”, “yellow”, “red” or “clear”. The colors “blue” and “purple” - although valid colors - should not be sent in a status-message, as these are handled specially by the Hobbit server.

The “additional text” normally includes a local timestamp and a summary of the test result on the first line. Any lines following the first one are free-form, and can include any information that may be useful to diagnose the problem being reported.

The LIFETIME defines how long this status is valid after being received by the Hobbit server. The default is 30 minutes, but you can set any period you like. E.g. for a custom test that runs once an hour, you will want

to set this to at least 60 minutes - otherwise the status will go purple after 30 minutes. It is a good idea to set the LIFETIME to slightly more than the interval between your tests, to allow for variations in the time it takes your test to complete. The LIFETIME is in minutes, unless you add an “h” (hours), “d” (days) or “w” (weeks) immediately after the number, e.g. “status+5h” for a status that is valid for 5 hours.

The GROUP option is used to direct alerts from the status to a specific group. It is currently used for status generated from the Hobbit clients’ data, e.g. to direct alerts for a “procs” status to different people, depending on exactly which process is down.

notify HOSTNAME.TESTNAME <message text> This triggers an informational message to be sent to those who receive alerts for this HOSTNAME+TESTNAME combination, according to the rules defined in *hobbit-alerts.cfg*(5). This is used by the *hobbit-enadis.cgi*(1) tool to notify people about hosts being disabled or enabled, but can also serve as a general way of notifying server administrators.

data HOSTNAME.DATANAME<newline><additional text> The “data” message allows tools to send data about a host, without it appearing as a column on the Hobbit webpages. This is used e.g. to report statistics about a host, e.g. vmstat data which does not in itself represent something that has a red, yellow or green identity. It is used by RRD bottom-feeder modules, among others. In Hobbit, data messages are by default processed only by the *hobbitd_rrd*(8) module. If you want to handle data-messages by an external application, you may want to enable the *hobbitd_filestore*(8) module for data-messages, to store data-messages in a format compatible with how the Big Brother daemon does.

Disables a specific test for DURATION minutes. This will cause the status of this test to be listed as “blue” on the BBDISPLAY server, and no alerts for this host/test will be generated. If DURATION is given as a number followed by s/m/h/d, it is interpreted as being in seconds/minutes/hours/days respectively. **To**

enable HOSTNAME.TESTNAME Re-enables a test that had been disabled.

query HOSTNAME.TESTNAME Query the BBDISPLAY server for the latest status reported for this particular test. If the host/test status is known, the response is the first line of the status report - the current color will be the first word on the line. Additional lines of text that might be present on the status-message cannot be retrieved.

This allows any Hobbit client to determine the status of a particular test, whether it is one pertaining to the host where the client is running, some other host, or perhaps the result of a combined test from multiple hosts managed by *bbcombotest*(1). This will typically be useful to Hobbit client extension scripts, that need to determine the status of other hosts e.g. to decide if an automatic recovery action should be initiated.

config FILENAME Retrieve one of the Hobbit configuration files from the server. This command allows a client to pull files from the \$BBHOME/etc/ directory on the server, allowing for semi-automatic updates of the client configuration. Since the configuration files are designed to have a common file for the configuration of all hosts in the system - and this is in fact the recommended way of configuring your clients - this makes it easier to keep the configuration files synchronized.

drop HOSTNAME Removes all data stored about the host HOSTNAME. It is assumed that you have already deleted the host from the bb-hosts configuration file.

drop HOSTNAME TESTNAME Remove data about a single test (column).

rename OLDHOSTNAME NEWHOSTNAME Rename all data for a host that changes its name. You should do this before changing the hostname in the bb-hosts configuration file.

rename HOSTNAME OLDTESTNAME NEWTESTNAME Rename data about a single test (column).

hobbitdlog HOSTNAME.TESTNAME Retrieve the Hobbit status-log for a single test. The first line of the response contain a series of fields separated by a pipe-sign:

hostname The name of the host

testname The name of the test

color Status color (green, yellow, red, blue, clear, purple)

testflags For network tests, the flags indicating details about the test (used by bbgen).

lastchange Unix timestamp when the status color last changed.

logtime Unix timestamp when the log message was received.

validtime Unix timestamp when the log message is no longer valid (it goes purple at this time).

acktime -1, or Unix timestamp when an active acknowledgement expires.

disabletime -1, or Unix timestamp when the status is no longer disabled.

sender IP-address where the status was received from.

cookie -1, or the cookie value used to acknowledge an alert.

ackmsg Empty, or the acknowledgment message sent when the status was acknowledged. Newline, pipe-signs and backslashes are escaped by with a backslash in C-style.

dismsg Empty, or the message sent when the status was disabled. Newline, pipe-signs and backslashes are escaped by with a backslash in C-style.

After the first line comes the full status log in plain text format.

hobbitdxlog HOSTNAME.TESTNAME Retrieves an XML-string with the status log as for the “hobbitdlog” command.

hobbitboard [CRITERIA [fields=FIELDLIST]] Retrieves a summary of the status of all known tests available to the Hobbit daemon.

By default - if no CRITERIA is provided - it returns one line for all status messages that are found in Hobbit. You can filter the response by selecting a page, a host, a test or a color - wildcards are not supported, so you can pick only one page, host, test or color.

page=PAGEPATH Include only tests from hosts found on the PAGEPATH page in the bb-hosts file.

host=HOSTNAME Include only tests from the host HOSTNAME

test=TESTNAME Include only tests with the testname TESTNAME

color=COLORNAME Include only tests where the status color is COLORNAME

You can filter on e.g. both a hostname and a testname.

The response is one line for each status that matches the CRITERIA, or all statuses if no criteria is specified. The line is composed of a number of fields, separated by a pipe-sign. You can select what fields to retrieve by listing them in the FIELDLIST. The following fields are available:

hostname The name of the host

testname The name of the test

color Status color (green, yellow, red, blue, clear, purple)

flags For network tests, the flags indicating details about the test (used by bbgen).

lastchange Unix timestamp when the status color last changed.

logtime Unix timestamp when the log message was received.

validtime Unix timestamp when the log message is no longer valid (it goes purple at this time).

acktime -1, or Unix timestamp when an active acknowledgement expires.

disabletime -1, or Unix timestamp when the status is no longer disabled.

sender IP-address where the status was received from.

cookie -1, or the cookie value used to acknowledge an alert.

line1 First line of status log.

ackmsg Empty (if no acknowledge is active), or the text of the acknowledge message.

dismsg Empty (if the status is currently enabled), or the text of the disable message.

msg The full text of the current status message.

The ackmsg, dismsg and msg fields have certain characters encoded: Newline is “\n”, TAB is “\t”, carriage return is “\r”, a pipe-sign is “\|”, and a backslash is “\\”.

If the “fields” parameter is omitted, a default set of hostname,testname,color,flags,lastchange,logtime,validtime,acktime,disabletime is used.

hobbitxboard Retrieves an XML-string with the summary of all status logs as for the “hobbitxboard” command.

download FILENAME Download a file from the Hobbit servers’ download directory.

client HOSTNAME.OSTYPE [HOSTCLASS] Used to send a “client” message to the Hobbit server. Client messages are generated by the Hobbit client; when sent to the Hobbit server they are matched against the rules in the *hobbit-clients.cfg(5)* configuration file, and status messages are generated for the client-side tests.

clientlog HOSTNAME [section=SECTIONNAME[,SECTIONNAME...]] Retrieves the current raw client message last sent by HOSTNAME. The optional “section” filter is used to select specific sections of the client data.

ping Attempts to contact the Hobbit server. If successful, the Hobbit server version ID is reported.

pullclient This message is used when fetching client data via the “pull” mechanism implemented by *hobbitfetch(8)* and *msgcache(8)* for clients that cannot connect directly to the Hobbit server.

ghostlist Report a list of **ghost** clients seen by the Hobbit server. Ghosts are systems that report data to the Hobbit server, but are not listed in the bb-hosts file.

schedule [TIMESTAMP COMMAND] Schedules an command sent to the Hobbit server for execution at a later time. E.g. used to schedule disabling of a host or service at sometime in the future. COMMAND is a complete Hobbit command such as the ones listed above. TIMESTAMP is the Unix epoch time when the command will be executed.

If no parameters are given, the currently scheduled tasks are listed in the response. The response is one line per scheduled command, with the job-id, the time when the command will be executed, the IP address from which this was sent, and the full command string. To cancel an previously scheduled command, “**schedule cancel JOBID**” can be used. JOBID is a number provided as the first item in the output from the schedule list.

9.2.5 EXAMPLE

Send a normal status message to the BBDISPLAY server, using the standard Hobbit protocol on TCP port 1984:

\$\$BB\$BBDISP”statuswww.foo.com.httpgreen‘date‘WebOK”

Send the same status message, but using HTTP protocol via the webservers’ bbmessage.cgi script:

\$\$BB`http://bb.foo.com/cgi-bin/bbmessage.cgi”statuswww.foo.com.httpgreen‘date‘WebOK”`

Use “query” message to determine the color of the “www” test, and restart Apache if it is red:

\$\$WWW='\$\$BB\$BBDISP”querywww.foo.com.www“—awk’{print\$1}‘‘ \$if[“\$WWW”=”red”];then/etc/init.d/apacherestart;fi

Use “config” message to update the local bb-dftab file (but only if we get a response):

\$\$BB\$BBDISP”configbb-dftab”>/tmp/bb-dftab.new \$if[-s/tmp/bb-dftab.new];then mv/tmp/bb-dftab.new\$BBHOME/etc/bb-dftab fi

Send a very large status message that has been built in the file “statusmsg.txt”. Instead of providing it on the command line, pass it via stdin to the bb command:

\$catstatusmsg.txt—\$\$BB\$BBDISP”@”

9.2.6 NOTES

This man-page describes the bb client program provided as part of Hobbit . This implementation provides features not present in the standard Big Brother bb client - specifically, the support for sending messages over HTTP, and many commands such as “query” and “config” are not part of the bb client shipped with Big Brother.

9.2.7 SEE ALSO

`bbcombotest(1)`, `bb-hosts(5)`, `hobbitserver.cfg(5)`, `hobbit(7)`

Chapter 10

Data Backend

10.1 HOBBITD_RRD

hobbitd_rrd - hobbitd worker module for updating Hobbit RRD files

10.1.1 SYNOPSIS

```
hobbitd_channel --channel=status hobbitd_rrd [options]
    hobbitd_channel --channel=data hobbitd_rrd [options]
```

10.1.2 DESCRIPTION

hobbitd_rrd is a worker module for hobbitd, and as such it is normally run via the *hobbitd_channel(8)* program. It receives “status” and “data” messages from hobbitd via stdin, and updates the RRD databases used to generate trend-graphs.

Clients can send data to Hobbit using both status- and data- messages. So you will normally run two instances of this module, once for the “status” channel and once for the “data” channel.

hobbitd_rrd understands data sent by the LARRD 0.43c client-side scripts (the so-called “bottom-feeder” scripts). So you still want to install the LARRD bottom-feeders on the clients you monitor.

Note: For certain types of data, the RRD files used by Hobbit are incompatible with those generated by the Big Brother LARRD add-on. See the COMPATIBILITY section below.

10.1.3 OPTIONS

-debug Enable debugging output.

-rrddir= DIRECTORY Defines the directory where the RRD-files are stored. hobbitd_rrd will use the location pointed to by the BBRRDS environment if this option is not present.

-extra-script= FILENAME Defines the script that is run to get the RRD data for tests that are not built into hobbitd_rrd. You must also specify which tests are handled by the external script in the **-extra-tests** option. This option can only be given once, so the script must handle all of the external test-data. See the CUSTOM RRD DATA section below. Note that this is NOT needed if your custom graphs are generated by the NCV (Name Colon Value) module described below, it is only required for data where you have a custom script to parse the status message and extract the data that is put into the graph.

-extra-tests= TEST[,TEST] List of testnames that are handled by the external script. See the CUSTOM RRD DATA section below. Note that NCV graphs should NOT be listed here, but in the TEST2RRD environment variable - see below.

-processor=COMMAND Feed the raw RRD data into COMMAND via standard input. COMMAND must be a simple command with no options; if necessary, COMMAND can be a script that runs the real command with the necessary options. The data sent to COMMAND consists of lines of text of the form

dsname1[:dsname2]time:value1[:value2]rrdtype[key1][key2] First the dataset names, corresponding to the DS definitions in the RRD files. Next the timestamp of the update, followed by the values in the same order as the dsname-list. Third item is the hostname, fourth is the type of RRD file. After that there may be between 0-2 additional keys: For “disk” this is the filesystem mountpoint, for “tcp” tests it is the service name, for “tcp” “http” tests there is also the URL checked. Note that filesystem names and URLs have forward slash substituted by comma. See the hobbitd/rrd/*.c sourcecode for details, look for the “setupfn” function calls. An example:

```
la 1196115478:7 myhost la
la 1196115478:169 myhost procs
la 1196115478:1 myhost users
la 1196115478:244 myhost clock
pct:used 1196115478:63:62338144 myhost disk ,root
realmempct 1196115478:76 myhost memory real
realmempct 1196115478:0 myhost memory swap
realmempct 1196115478:20 myhost memory actual
sec 1196115478:0.03 tcp myhost conn
sec 1196115478:1.38 myhost tcp http www.hobbitmon.com,
```

10.1.4 ENVIRONMENT

TEST2RRD Defines the mapping between a status-log columnname and the corresponding RRD database format. This is normally defined in the *hobbitserver:cfg(5)* file.

BRRRDS Default directory where RRD files are stored.

NCV_testname Defines the types of data collected by the “ncv” module in hobbitd_rrd. See below for more information.

10.1.5 COLLECTED DATA

The following RRD-file datasets are generated by hobbitd_rrd:

la Records the CPU load average. Data is collected from the “cpu” status report. Requires that a Hobbit client is running on the monitored server.

disk Records the disk utilization. Data is collected from the “disk” status report. Requires that a Hobbit-compatible client is running on the monitored server.

memory Records memory- and swap-utilization. Data is collected from the “memory” status report. If no “memory” status is reported, it will use the data from the Win32 client “cpu” status report to generate this dataset. Requires that a Hobbit-compatible client is running on the monitored server.

netstat Records TCP and UDP statistics. Data is collected from the “netstat” status report; however, this data is often sent via the Hobbit “data” protocol, so there need not be a “netstat” column visible on the Hobbit display. To get these data, the LARRD netstat bottom-feeder script must be running on the monitored server.

vmstat Records system performance metrics from the “vmstat” command. Data is collected from the “vmstat” status report; however, this data is often sent via the Hobbit “data” protocol, so there need not be a “vmstat” column visible on the Hobbit display. To get these data, the LARRD vmstat bottom-feeder script must be running on the monitored server.

tcp Response-time metrics from all of the Hobbit network tests are recorded in the “tcp” RRD.

apache Apache server performance metrics, taken from the “apache” data report. See the description of the **apache** keyword in *bb-hosts(5)* for details.

sendmail Sendmail server performance metrics, taken from the “mailstats” output. To get these data, the LARRD sendmail bottom-feeder script must be running on the monitored server.

mailq Mail queue size. To get these data, the LARRD nmailq bottom-feeder script must be running on the monitored server.

bea BEA Weblogic performance data. This is an experimental set of data collected from BEA Weblogic servers via SNMP, by the “beastats” tool included with Hobbit.

iishealth IIS webserver performance data, collected by the “iishealth” script. This script is a client-side add-on available from the www.deadcat.net archive.

temperature Temperature data, collected with the temperature script from www.deadcat.net. To get these data, the temperature script must be running on the monitored server.

ntpstat Tracks the deviation between the local system time and an NTP server, using the output from the “ntpq -c rv” command. A simple script to collect these data is included in the Hobbit contrib/ directory.

citrix Tracks the number of active sessions on a Citrix server using the “query session” command. An extension for the BBNT client that generates data for this graph is in the Hobbit contrib/ directory.

10.1.6 CUSTOM RRD DATA IN NAME-COLON-VALUE (NCV) FORMAT

Many data-collection scripts report data in the form “NAME : value” or “NAME = value”. So a generic module in *hobbitd_rrd* allows for easy tracking of this type of data.

The “ncv” module will automatically detect all occurrences of a “NAME : value” or “NAME = value” string in a status message, and generate an RRD file holding all of the name/value data found in the message. The colon- or equal-sign must be present - if there is only whitespace, this module will fail.

Only the valid letters (A-Z, a-z) and digits (0-9) are used in the dataset names; whitespace and other characters are stripped off automatically. Only the first 19 characters of a dataset name are used (this is an RRD limitation). Underscore ‘_’ is not allowed, even though RRDtool permits this, and will be stripped from the name.

Note that each “NAME : value” must be on a line by itself. If you have a custom script generating the status- or data-message that is fed into the NCV handler, make sure it inserts a newline before each of the data-items you want to track.

To enable the ncv module for a status, add a “COLUMNNAME=ncv” to the TEST2RRD setting and the COLUMNNAME to the GRAPHS setting in *hobbitserver.cfg(5)*, then restart Hobbit. Hobbit will now send all status-messages for the column COLUMNNAME through the *hobbitd_rrd* ncv-handler.

The name of the RRD file will be COLUMNNAME.rrd.

By default, all of the datasets are generated as the RRD type “DERIVE” which works for all types of monotonically increasing counters. If you have data that are of the type GAUGE, you can override the default via an environment variable NCV_COLUMNNAME.

E.g. if you are using the bb-mysqlstatus script from www.deadcat.net to collect data about your MySQL server, it generates a report in the column called “mysql”. One data item is the average number of queries/second, which must be logged in the RRD file as type “GAUGE”. To do that, add the following to *hobbitserver.cfg*:

NCV_mysql=”Queriespersecondavg:GAUGE” If you have multiple datasets that you myst define, add them to the environment variable separated by commas, e.g.

NCV_mysql=”Uptime:NONE,Queriespersecondavg:GAUGE”

The dataset type “NONE” used above causes *hobbitd_rrd* to ignore this data, it is not included in the RRD file.

You can use “*” as the dataset name to match all datasets not listed. E.g.

`NCV_weather="Rain:DERIVE,:GAUGE"` will cause the “Rainfall” dataset to be of type DERIVE, and all others of type GAUGE. If you want to track only a few of the variables in your data, you can use “*:NONE” to drop any dataset not explicitly listed.

For a more detailed “how to” description, see the on-line HTML documentation of “How to create graph custom data” available in the Help menu section on your Hobbit server.

10.1.7 CUSTOM RRD DATA VIA SCRIPTS

`hobbitd_rrd` provides a simple mechanism for adding custom graphs to the set of data collected on your Hobbit server. By adding the “`-extra-script`” and “`-extra-tests`” options, data reported to Hobbit from selected tests are passed to an external script, which can define the RRD data-sets to store in an RRD file.

NOTE: For performance reasons, you should not use this mechanism for large amounts of data. The overhead involved in storing the received message to disk and launching the script is significantly larger than the normal `hobbitd_rrd` overhead. So if you have a large number of reports for a given test, you should consider implementing it in C and including it in the `hobbitd_rrd` tool.

Apart from writing the script, You must also add a section to `hobbitgraph.cfg(5)` so that `hobbitgraph.cgi(1)` knows how to generate the graph from the data stored in the RRD file. To make the graphs actually show up on the status-page and/or the “trends” page, add the name of the new graph to the TEST2RRD and/or GRAPHS setting in `hobbit-server.cfg(5)`.

The script is invoked for each message that arrives, where the test-name matches one of the testnames given in the “`-extra-tests`” option. The script receives three commandline parameters:

Hostname The name of the host reporting the data.

Testname The name of the test being reported.

Filename File containing the data that was reported. This file is generated for you by `hobbitd_rrd`, and is also deleted automatically after your script is finished with it.

The script must process the data that is reported, and generate the following output:

RRD data-set definitions For each dataset that the RRD file holds, a line beginning with “DS:” must be output.

If multiple data-sets are used, print one line for each dataset. Data-set definitions are described in the `rrd-create(1)` documentation, but a common definition for e.g. tracking the number of users logged on would be “`DS:users:GAUGE:600:0:U`”. “users” is the name of the dataset, “GAUGE” is the datatype, “600” is the longest time allowed between updates for the data to be valid, “0” is the minimum value, and “U” is the maximum value (a “U” means “unknown”).

RRD filename The name of the RRD file where the data is stored. Note that Hobbit stores all RRD files in host-specific directories, so unlike LARRD you should not include the hostname in the name of the RRD file.

RRD values One line, with all of the data values collected by the script. Data-items are colon-delimited and must appear in the same sequence as your data-set definitions, e.g. if your RRD has two datasets with the values “5” and “0.4” respectively, then the script must output “`5:0.4`” as the RRD values. In some cases it may be useful to define a dataset even though you will not always have data for it. In that case, use “U” (unknown) for the value.

If you want to store the data in multiple RRD files, the script can just print out more sequences of data-set definitions, RRD filenames and RRD values. If the data-set definitions are identical to the previous definition, you need not print the data-set definitions again - just print a new RRD filename and value.

The following sample script for tracking weather data shows how to use this mechanism. It assumes the status message include lines like these:

```
green Weather in Copenhagen is FAIR
```

```
Temperature: 21 degrees Celsius
Wind: 4 m/s
Humidity: 72 %
Rainfall: 5 mm since 6:00 AM
```

A shell-script to track all of these variables could be written like this:

```
#!/bin/sh

# Input parameters: Hostname, testname (column), and messagefile
HOSTNAME="$1"
TESTNAME="$2"
FNAME="$3"

if [ "$TESTNAME" = "weather" ]
then
    # Analyze the message we got
    TEMP='grep "^\$Temperature:' $FNAME | awk '{print $2}''
    WIND='grep "^\$Wind:' $FNAME | awk '{print $2}''
    HMTY='grep "^\$Humidity:' $FNAME | awk '{print $2}''
    RAIN='grep "^\$Rainfall:' $FNAME | awk '{print $2}''

    # The RRD dataset definitions
    echo "DS:temperature:GAUGE:600:-30:50"
    echo "DS:wind:GAUGE:600:0:U"
    echo "DS:humidity:GAUGE:600:0:100"
    echo "DS:rainfall:DERIVE:600:0:100"

    # The filename
    echo "weather.rrd"

    # The data
    echo "$TEMP:$WIND:$HMTY:$RAIN"
fi

exit 0
```

10.1.8 COMPATIBILITY

Some of the RRD files generated by hobbitd_rrd are incompatible with the files generated by the Big Brother LARRD add-on:

vmstat The vmstat files with data from Linux based systems are incompatible due to the addition of a number of new data-items that LARRD 0.43 do not collect, but hobbitd_rrd does. This is due to changes in the output from the Linux vmstat command, and changes in the way e.g. system load metrics are reported.

netstat All netstat files from LARRD 0.43 are incompatible with hobbitd_rrd. The netstat data collected by LARRD is quite confusing: For some types of systems LARRD collects packet-counts, for others it collects byte- counts.

hobbitd_rrd uses a different RRD file-format with separate counters for packets and bytes and tracks whatever data the system is reporting.

10.1.9 SEE ALSO

`hobbitd_channel(8)`, `hobbitd(8)`, `hobbitserver.cfg(5)`, `hobbit(7)`

Chapter 11

Miscellaneous programs

11.1 HOBBITLAUNCH

`hobbitlaunch` - Master program to launch other Hobbit programs

11.1.1 SYNOPSIS

`hobbitlaunch [options]`

11.1.2 DESCRIPTION

`hobbitlaunch(8)` is the main program that controls the execution and scheduling of all of the components in the Hobbit system.

`hobbitlaunch` allows the administrator to add, remove or change the set of Hobbit applications and extensions without restarting Hobbit - `hobbitlaunch` will automatically notice any changes in the set of tasks, and change the scheduling of activities accordingly.

`hobbitlaunch` also allows the administrator to setup specific logfiles for each component of the Hobbit system, instead of getting output from all components logged to a single file.

11.1.3 OPTIONS

-env=FILENAME Loads the environment from FILENAME before starting other tools. The environment defined by FILENAME is the default, it can be overridden by the ENVFILE option in `hobbitlaunch.cfg(5)`

-config=FILENAME This option defines the file that `hobbitlaunch` scans for tasks it must launch. A description of this file is in `hobbitlaunch.cfg(5)`

The default tasklist is `/etc/hobbitlaunch.cfg`

-log=FILENAME Defines the logfile where `hobbitlaunch` logs information about failures to launch tasks and other data about the operation of `hobbitlaunch`. Logs from individual tasks are defined in the `hobbitlaunch.cfg` file. By default this is logged to `stdout`.

-pidfile=FILENAME Filename which `hobbitlaunch` saves its own process-ID to. Commonly used by automated start/stop scripts.

-verbose Logs the launch of all tasks to the logfile. Note that the logfile may become quite large if you enable this.

-dump Just dump the contents of the `hobbitlaunch.cfg` file after parsing it. Used for debugging.

-debug Enable debugging output while running.

-no-daemon `hobbitlaunch` normally detaches from the controlling tty and runs as a background task. This option keeps it running in the foreground.

11.1.4 STARTING TASKS

`hobbitlaunch` will read the configuration file and start all of the tasks listed there.

If a task completes abnormally (i.e. terminated by a signal or with a non-zero exit status), then `hobbitlaunch` will attempt to restart it 5 times. If it still will not run, then the task is disabled for 10 minutes. This will be logged to the `hobbitlaunch` logfile.

If the configuration file changes, `hobbitlaunch` will re-read it and notice any changes. If a running task was removed from the configuration, then the task is stopped. If a new task was added, it will be started. If the command used for a task changed, or it was given a new environment definition file, or the logfile was changed, then the task is stopped and restarted with the new definition.

11.1.5 SEE ALSO

`hobbitlaunch.cfg(5)`, `hobbit(7)`

Appendix A

Making of this book

A.1 Using Mercurial's book template to build our bookor

This book is converted into latex version from troff version Hobbit manpages.

It has a Makefile to automate the process book making.

1. Get hobbit manpages.
2. rtf2latex
3. fix latex syntax issue.
4. Prepare Diagram in InkScape.
5. Indexes
6. Table of contents
7. References
8. List of figures
9. List of Tables
10. Indexes
11. Download a recent source tarball from <http://www.selenic.com/mercurial/download>.
12. Unpack the tarball:

```
1 gzip -dc mercurial-version.tar.gz | tar xf -
```

13. Go into the source directory and run the installer script. This will build Mercurial and install it in your home directory.

```
1 cd mercurial-version
2 python setup.py install --force --home=$HOME
```

Once the install finishes, Mercurial will be in the `bin` subdirectory of your home directory. Don't forget to make sure that this directory is present in your shell's search path.

You will probably need to set the `PYTHONPATH` environment variable so that the Mercurial executable can find the rest of the Mercurial packages. For example, on my laptop, I have set it to `/home/bos/lib/python`. The exact path that you will need to use depends on how Python was built for your system, but should be easy to figure out. If you're uncertain, look through the output of the installer script above, and see where the contents of the `mercurial` directory were installed to.

A.2 On Windows

Building and installing Mercurial on Windows requires a variety of tools, a fair amount of technical knowledge, and considerable patience. I very much *do not recommend* this route if you are a “casual user”. Unless you intend to hack on Mercurial, I strongly suggest that you use a binary package instead.

If you are intent on building Mercurial from source on Windows, follow the “hard way” directions on the Mercurial wiki at <http://www.selenic.com/mercurial/wiki/index.cgi/WindowsInstall>, and expect the process to involve a lot of fiddly work.

Appendix B

Open Publication License

Version 1.0, 8 June 1999

B.1 Requirements on both unmodified and modified versions

The Open Publication works may be reproduced and distributed in whole or in part, in any medium physical or electronic, provided that the terms of this license are adhered to, and that this license or an incorporation of it by reference (with any options elected by the author(s) and/or publisher) is displayed in the reproduction.

Proper form for an incorporation by reference is as follows:

Copyright (c) *year* by *author's name or designee*. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, vx.y or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

The reference must be immediately followed with any options elected by the author(s) and/or publisher of the document (see section [B.6](#)).

Commercial redistribution of Open Publication-licensed material is permitted.

Any publication in standard (paper) book form shall require the citation of the original publisher and author. The publisher and author's names shall appear on all outer surfaces of the book. On all outer surfaces of the book the original publisher's name shall be as large as the title of the work and cited as possessive with respect to the title.

B.2 Copyright

The copyright to each Open Publication is owned by its author(s) or designee.

B.3 Scope of license

The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document.

Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

Severability. If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

No warranty. Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement.

B.4 Requirements on modified works

All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements:

1. The modified version must be labeled as such.
2. The person making the modifications must be identified and the modifications dated.
3. Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices.
4. The location of the original unmodified document must be identified.
5. The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

B.5 Good-practice recommendations

In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

1. If you are distributing Open Publication works on hardcopy or CD-ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.
2. All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.
3. Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD-ROM expression of an Open Publication-licensed work to its author(s).

B.6 License options

The author(s) and/or publisher of an Open Publication-licensed document may elect certain options by appending language to the reference to or copy of the license. These options are considered part of the license instance and must be included with the license (or its incorporation by reference) in derived works.

- A To prohibit distribution of substantively modified versions without the explicit permission of the author(s). “Substantive modification” is defined as a change to the semantic content of the document, and excludes mere changes in format or typographical corrections.

To accomplish this, add the phrase “Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.” to the license reference or copy.

- B To prohibit any publication of this work or derivative works in whole or in part in standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

To accomplish this, add the phrase “Distribution of the work or derivative of the work in any standard (paper) book form is prohibited unless prior permission is obtained from the copyright holder.” to the license reference or copy.

Bibliography

[Wau] Tim Waugh. patchutils—programs that operate on patch files. <http://cyberelk.net/tim/patchutils/>.

[Yan] T.J Yang. System monitoring with hobbit. http://en.wikibooks.org/wiki/System_Monitoring_with_Hobbit.

Index

PYTHONPATH environment variable, [79](#)
bb-hosts command, [13](#)
bbcmand command, [40](#)
bbcbotest.cfg command, [42](#), [45](#), [46](#)
bb command, [64](#)
hobbit-alerts.cfg command, [13](#)
hobbit-clients.cfg command, [13](#)
hobbitgraph.cfg command, [13](#), [43](#), [47](#)
hobbitlaunch.cfg command, [13](#), [42](#), [56](#)
hobbitserver.cfg command, [13](#)
hostinfo.csv command, [51](#)

environment variables
 PYTHONPATH, [79](#)