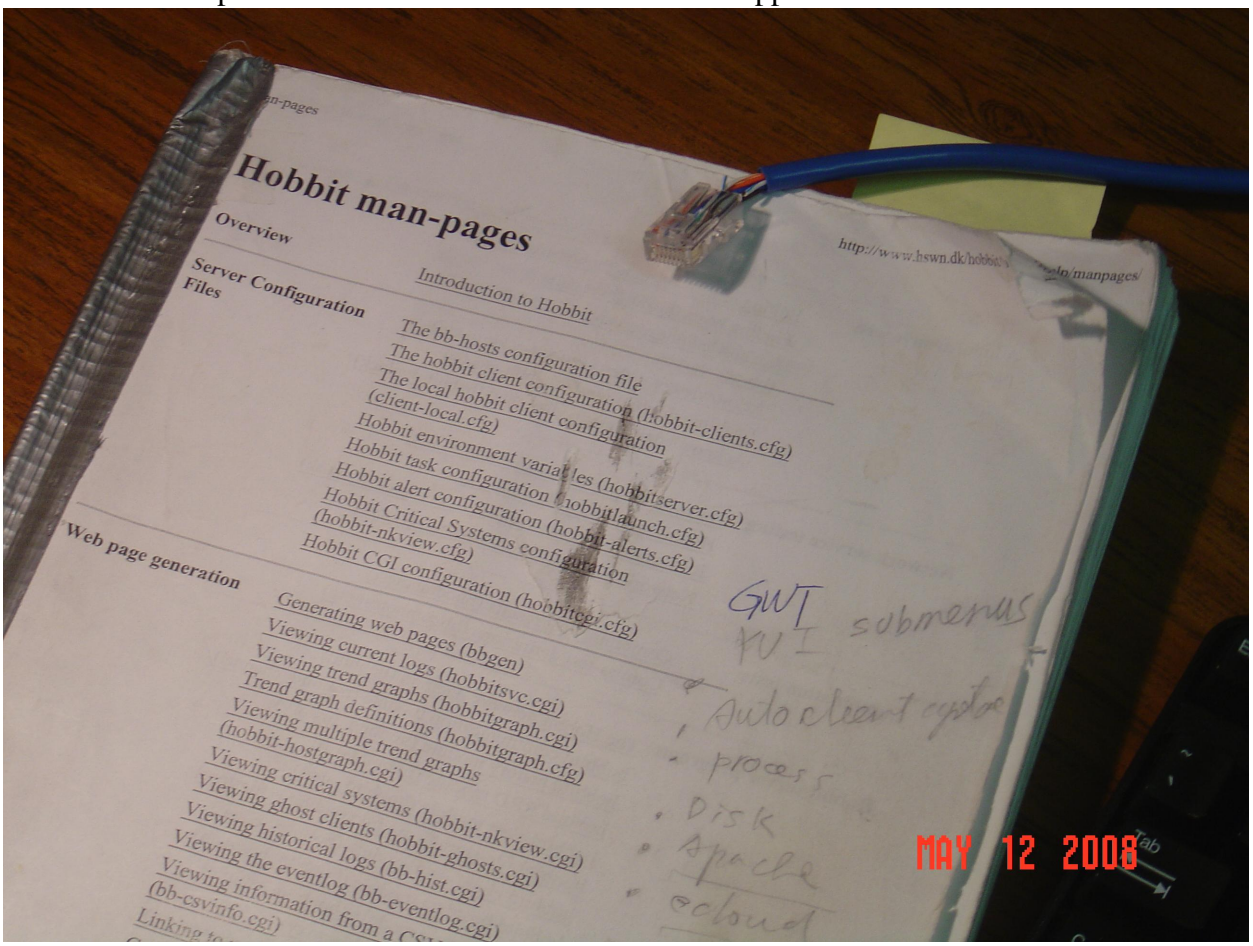
Hobbitmon Developer Guide

Hobbitmon Community



Contents

List of Figures

List of Tables

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 orignial 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 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 page, 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 alredy 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 wit 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 relased 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

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. bbcmd(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

Hobbitmon modules development

3.1 module programming

This describes how to setup a Hobbit server for monitoring your systems. It assumes that you are setting up a full Hobbit server - i.e. either you do not have a Big Brother server, or you will replace it completely with Hobbit.

Note to Big Brother users: Although some of the Hobbit tools have evolved from the bbgen toolkit that was used on top of a Big Brother server installation, the Hobbit versions of these tools now require that you run Hobbit - not Big Brother. If you are migrating from Big Brother to Hobbit, then you should follow the migration guide.

3.1.1 Prerequisites - before you install Hobbit

There are a few things you should check before you begin to install Hobbit. Dont be scared of the number of items here - it is likely that you already have most or all of it in place. **A webbrowser capable of handling HTML 4, JavaScript and CSS**

This includes most browsers available today - Internet Explorer 5 or later, all Mozilla/Firefox versions, Konqueror, Netscape 6 and several others. The old Netscape 4.x browsers are known NOT to work. **A Unix-like operating system**

Hobbit is written for Unix-based systems, e.g. Linux, FreeBSD, or Solaris. It will probably work on any Unix-like system that supports the Unix System V IPC mechanisms (shared memory, semaphores) - that should be just about anything Unix-like you are likely to have. **Sufficient SYSV IPC ressources on your system**

Hobbit uses 8 shared memory segments, ranging in size from 32 KB to 512 KB (2336 KB total) in the default configuration; and 8 sets of 3 semaphores. Experience shows that some systems need tuning to provide the necessary IPC ressources that Hobbit uses. Specifically, when installing on Solaris you must increase the "shmseg" kernel parameter from the default 6 to at least 8. Since other programs on your system may also use shared memory, a higher value may be required. See http://www.hswn.dk/hobbiton/2005/08/msg00183.html for more information about these issues. **A webserver**

Hobbit is designed with a web-based front-end. So you should have a webserver such as Apache running on the server where you install Hobbit. **A working C compiler, GNU make.**

Hobbit is written in C, so you need a working C compiler, e.g. gcc. You will also need a "make" utility - many systems have one by default, but you need to use the GNU make utility. On some systems, this is pre-installed as "gmake" or "gnumake". The configure-script checks this for you.

HP-UX users should note that the HP-supplied C compiler is known to mis-compile the lib/environ.c file, and produces an output file lib/environ.o of length 0 bytes. HP-UX users on the hobbit mailing list agree that the default C compiler shipped with HP-UX should not be used to compile Hobbit - it is only for re-building the HP-UX kernel. The GNU C compiler works fine on HP-UX. More details in this e-mail from the Hobbit mailing list. **PCRE, RRDtool, libpng, OpenSSL, OpenLDAP, Net-SNMP libraries.**

Hobbit relies on a number of Open-Source libraries - these must be installed before you start building Hobbit. On many systems you already have these pre-installed - they are commonly installed by default on Linux systems, and FreeBSD has all of them in the "ports" collection.

Note: Although many systems have these libraries pre-installed, they often include only the run-time libraries and not the files that are needed to compile and build programs such as Hobbit. So if you think you have all of these libraries installed but Hobbit will not build, do check that you have the development files installed as well. Often these are in packages called "something-*dev* ".

- PCRE - Perl Compatible Regular Expression library - is a library for matching text-strings. It is available from http://www.pcre.org/
- RRDtool is a library for handling the Round-Robin Databases used to hold the historical data Hobbit gathers. It is available from http://oss.oetiker.ch/rrdtool/. Hobbit is known to work with RRDtool 1.0.x - if you prefer to use the newer RRDtool 1.2.x, make sure you use at least version 1.2.2.
- libpng is a library for generating images in the PNG format. It is used by RRDtool (and hence Hobbit). You can find it at http://www.libpng.org/pub/png/libpng.html
- OpenSSL is a library for communicating with network services, that use SSL encryption - e.g. secure websites. Although this library is not absolutely required for Hobbit, I strongly recommend that you install it because sooner or later you will probably need it anyway. It is available from http://www.openssl.org/. Note: If you are building on Solaris, you should check that you have a random-data generator, either the prngd daemon (available on Sun Freeware) or the Solaris /dev/random driver from Solaris patch 112438.
- OpenLDAP is used to query LDAP directory servers. If you would like to test that your directory server is up and running, you will need this library. It is available from http://www.openldap.org/
- Net-SNMP is used to query SNMP-based systems, typically network devices like routers, switches or firewalls. But nearly all types of computer systems have some sort of SNMP support. If you would like to collect data from SNMP-enabled systems you will need this library. It is available from http://net-snmp.sourceforge.net/

The configure-script will attempt to locate all of these libraries on your system, and complain if the required ones are missing. **A "hobbit" userid on your system**

A core element of Hobbit is a network daemon. To keep your system secure and limit the amount of damage that can be done if someone finds a security problem in Hobbit, I strongly recommend that you create a dedicated userid for the Hobbit programs. This user should *not* be a member of any other groups on your system.

Hobbit will install the hobbitping tool as **setuid-root** (only on the Hobbit server). This program requires root privileges to be able to perform network "ping" tests. It will drop root privileges immediately after obtaining the network socket needed for this, and will not run with root privileges at all while handling network traffic or doing file I/O.

3.1.2 Building Hobbit

After unpacking Hobbit from the tar-file, run the configure script. This script asks a series of questions, but all of the questions have a reasonable default response. So if you are in doubt about what to answer, use the default setting. You can see what it looks like.

When the configure script finishes, it tells you to run make to build the Hobbit programs. If your default "make" tool is not GNU make, you should use the command for running GNU make instead, e.g. gmake. You will now see a lot of commands being run to build the programs, it usually takes a minute or two.

When it is finished, you finish the installation by running make install.

The first time you run make install, besides installing the Hobbit programs it also creates the default directory structure used by Hobbit, and installs an initial set of configuration files that you can use as the basis for setting up monitoring of your entire network.

It is safe to run make install when upgrading a Hobbit server. It installs the programs, adds new template-files that were not present in your previous version, and updates your configuration files with any new sections that have been added. Any changes you have made yourself are preserved.

3.1.3 Configuring your webserver

Hobbit uses a web-based front-end. So you need to configure your webserver so that it knows where the Hobbit webpages can be found, and what CGI scripts can run as part of Hobbit. This usually means adding a few lines to your webserver configuration that sets up a URL which points at the */server/www/* directory, and which tells your webserver that the */cgi-bin/* directory holds CGI scripts that the webserver should run when they are requested.

If you are using the Apache webserver, you will find the necessary additions to the Apache configuration in */server/etc/hobbit-apache.conf* - it looks like this. After changing the webserver configuration, you probably need to restart the webserver.

If you configured Hobbit to put the Administration CGI scripts into a separate directory (recommended for better security), you will also need to setup the password-file that controls access to this directory. Use the htpasswd command both to create the password file and to add or delete users:

```
# /usr/sbin/htpasswd -c /usr/local/hobbit/server/etc/hobbitpasswd admin
New password:
Re-type new password:
Adding password for user admin
#
```

The **-c** option should only be used the first time, to create the password file. See the Apache documentation for details about how to use htpasswd.

3.1.4 Starting Hobbit

You can now login as the "hobbit" user, and run the command ./server/hobbit.sh start to start Hobbit. After a few seconds, it should have started and you now have the following processes running:

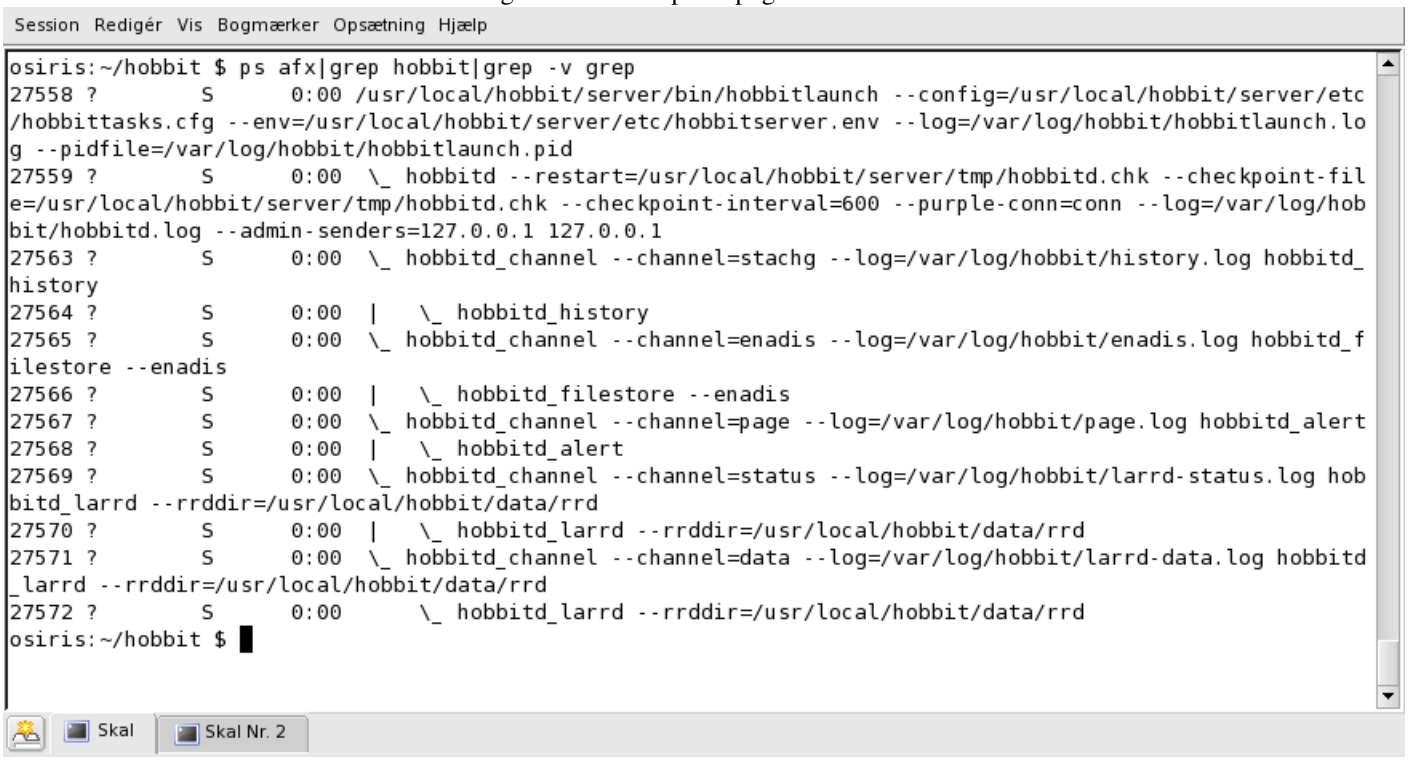
Quite a few, but all of them controlled by the master hobbitlaunch process. A quick run-down of what each of them does:

- hobbitd is the network daemon that receives status updates from the clients and the network test tool. It also provides the current status of all your systems to the tool that generates the webpages.
- hobbitd_channel provides the communication between hobbitd and all of the helper modules that implement other server-based functions.
- hobbitd_history takes care of recording the history of status changes for each item you monitor. This is used to track what has happened with a single status over time - when it was red, when it was green, what the error reported at 2:51 AM last Friday looked like. The history file format is compatible with the format used by the Big Brother package.
- hobbitd_filestore stores files with information about the current status of the systems monitored by Hobbit. There may be several of these running, but normally you will only need the one that stores information about hosts that have been disabled, which is the one you see here.
- hobbitd_alert takes care of sending out alerts when your servers begin to report a critical status.
- hobbitd_rrd updates the RRD database files with the numeric data collected from the status reports, to track e.g. how the disk utilization of a server changes over time. There are two of these processes, because the data can arrive in two different ways.

After a couple of minutes, you should have data available for the Hobbit server itself. If you open a webbrowser with the Hobbit URL - usually http://your.server/hobbit/ - you should see something like this:

Each of the little faces indicate an item that is being monitored for this host. Here you see the default set of items that the Hobbit installation sets up for a Hobbit server:

Figure 3.1: hobbitprocs.png



- *bbd* is the availability of the Hobbit network daemon.
- *bbgen* is the status of the bbgen tool, which updates the webpages.
- *bbtest* is the status of the bbtest-net network tester that performs all of the network tests you configure in Hobbit.
- *conn* is a simple "ping" test of the host.
- *hobbitd* is the status of the Hobbit daemon, with statistics about how many monitored items are being tracked.
- *http* is the status of the HTTP-server running on the Hobbit server.
- *info* contains information about how the host is configured in Hobbit, such as what IP-address it has, what network tests are being run against this host etc.
- *trends* is a collection of the various RRD graphs available for this host.

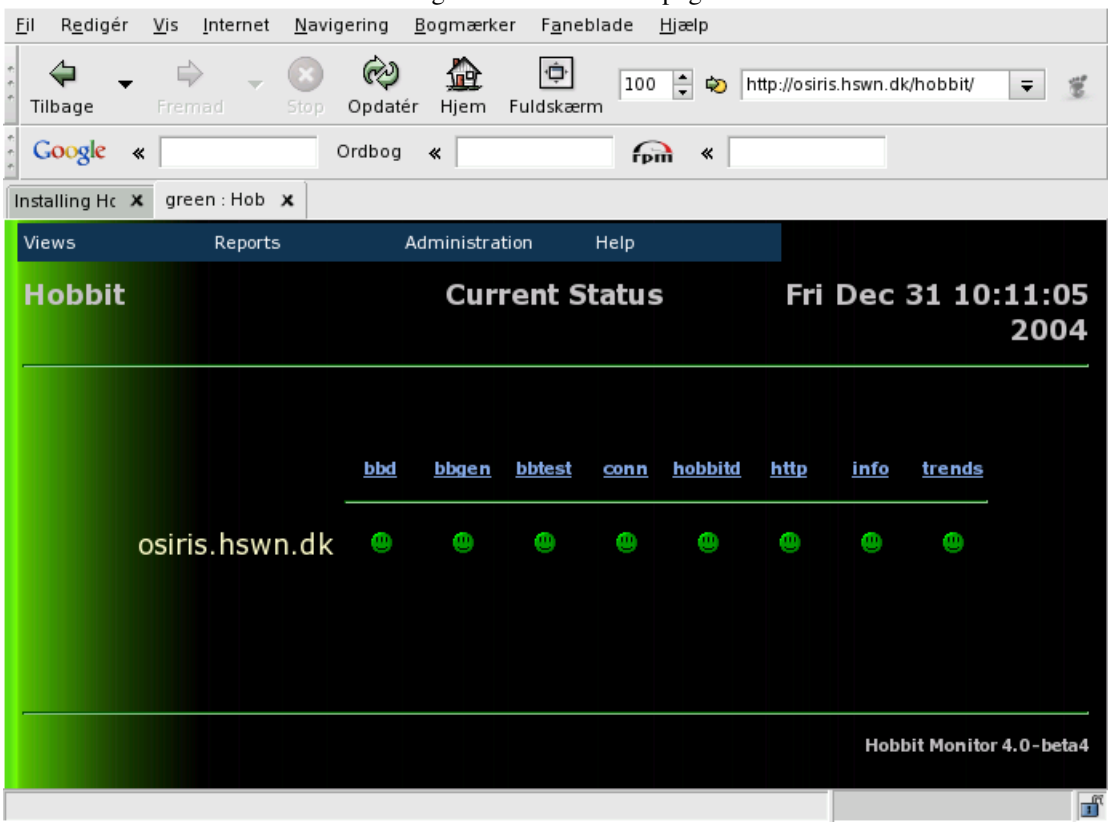
You can click on each of the green icons to see a more detailed status.

3.1.5 Next steps

Congratulations, you now have a running Hobbit server!

The next step is to configure it to monitor your servers and applications, and to set up the alerts to send you e-mail, call a pager, or send an SMS in case of trouble. For that, see the Hobbit configuration guide.

Figure 3.2: hobbitmain.png



Chapter 4

Miscellanceours programs

4.1 HOBBITLAUNCH

hobbitlaunch - Master program to launch other Hobbit programs

4.1.1 SYNOPSIS

hobbitlaunch [options]

4.1.2 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.

4.1.3 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.

4.1.4 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.

4.1.5 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.

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

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:

```
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.

```
cd mercurial-version
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, v*x.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