

Mars 2.2.7 User's Guide

Brian Trammell
brian@lfrd.com

Leapfrog Research and Development, LLC

March 22, 2004

Abstract

Describes the Mars network status monitor, and walks a new user through installation, configuration, and use.

Contents

1	About Mars	2
2	The Mars Distribution	2
3	Installing Mars	3
3.1	System Requirements	3
3.2	Distribution Contents	3
4	Starting Mars	4
5	Adding Hosts and Services	5
5.1	The Host Editor	5
5.2	The Service Editor	6
5.3	Keyboard Editing	8
6	Monitoring Services with Mars	8
7	Plugins and Probe Extensions	11
7.1	UI Notification	11
7.2	Mail Notification	12
7.3	CSV Logger	12
7.4	XML Snapshot	13
7.5	XMPP Notification	13
7.6	Client Debugger	14
7.7	Mac OS X Integration	14
7.8	HTTPS Probe Extension	14
7.9	JDBC Probe Extension	15

8	Mars on Mac OS X	15
A	Change History	16
A.1	Mars 2.2.7 – March 22, 2004	16
A.2	Mars 2.2.6 – March 1, 2004	16
A.3	Mars 2.2.5 – February 10, 2004	17
A.4	Mars 2.2.4 – February 4, 2004	17
A.5	Mars 2.2.3 – January 26, 2004	18
A.6	Mars 2.2.2 – March 11, 2003	18
A.7	Mars 2.2.1 – February 25, 2003	19
A.8	Mars 2.2.0 – October 28, 2002	19
A.9	Mars 2.1.3 – August 12, 2002	20
A.10	Mars 2.1.2 – July 14, 2002	20
A.11	Mars 2.1.1 – June 30, 2002	20
A.12	Mars 2.1.0 – June 5, 2002	20
A.13	Mars 2.0.0 – May 8, 2002	21

1 About Mars

Mars is a simple services-oriented network status monitor written in Java. It monitors a network by simulating client connections to Internet services and reporting when those services are not responding. It is quick and easy to install and configure, which distinguishes it from other, more complex network monitoring tools.

2 The Mars Distribution

The Mars distribution is available at <http://www.altara.org/mars.html>. Mars is released under the GNU General Public License, version 2. This means you are free to modify and redistribute it, provided that you make your modifications available under the GPL. The text of the GPL is available in the `COPYING` file in the `doc` directory in the Mars distribution for details.

The Mars distribution contains the following open-source software; the licenses for these packages are found in the `doc` directory in the Mars distribution. We thank the developers of these packages for making them available.

- GNU Getopt version 1.0.9, by Aaron Renn.
- JDOM beta 9, ©2000-2003 Brett McLaughlin and Jason Hunter, all rights reserved. JDOM is developed by the JDOM Project at <http://www.jdom.org/>.
- Apache Xerces-J 2.6.0, ©1999-2003 The Apache Software Foundation, all rights reserved. Xerces is developed by the Apache Software Foundation at <http://www.apache.org/>.

- Apache Jakarta-Oro 2.0.8, ©1999-2003 The Apache Software Foundation, all rights reserved. Jakarta-Oro is developed by the Apache Software Foundation at <http://www.apache.org/>.

In addition, Mars is packaged with several probe extensions and plugins, in the `extras` directory.. All of these extensions are released under the GNU General Public License, version 2. All are ©2003-2004 Leapfrog Research and Development, LLC, except the following:

- HTTPS Probe Extension, ©2003 Scott Ahten.

3 Installing Mars

The Mars distribution is self-contained; simply move the distribution directory to a convenient place and you're ready to go. This is known as the Mars "home" directory. To use optional extensions shipped in the `extras` directory, simply move them to the home directory. Though Mars is designed to be run from its home directory, this behavior can be changed with the `--home` command-line switch described below.

Mac OS X users: see section 8 for information on installing Mars on Mac OS X.

3.1 System Requirements

Mars is a Java application, and therefore requires a Java runtime environment to be installed on your machine before running. Any J2SE runtime version 1.3 or later should do. Mars 2.2.5 has been tested to run on the following Java runtime environments:

- Sun JRE 1.4.2-03 for Linux/i586
- Java 1.4.1-01 for Mac OS X 10.3 (Panther)

Mars also requires an XML parser. Java 1.4 includes an XML parser, and the Mars distribution includes Apache Xerces for running on Java 1.3, so this isn't really a requirement you need to worry about.

3.2 Distribution Contents

The Mars core is distributed in two versions, `mars.jar` for Java 1.4. and `mars-j13.jar` for Java 1.3. The only difference between the two versions is that the former contains an XML parser (Apache Xerces), which is included with the JRE on Java 1.4. The distribution directory contains only the former `jar` file; `mars-j13.jar` is available as a separate download.

The distribution directory contains the probe definition file `mars-def.xml`, which defines the protocols for text-based services Mars knows about. See the *Hacker's Guide* for the structure of this file.

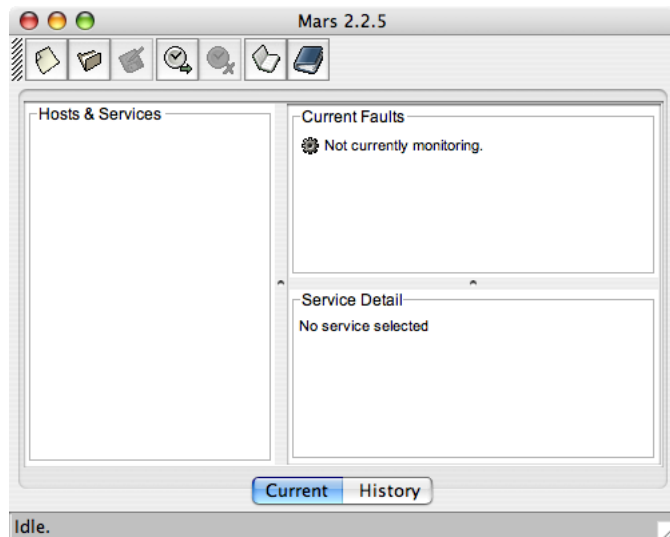


Figure 1: The Mars main window with a new, empty configuration

The distribution directory also plugin and probe extension classes distributed with Mars in the `extras` directory. *Note for users of Mars 2.2.5 and earlier versions: all plugins and probe extensions have been moved to `extras`; you'll need to move these to the Mars home directory to use them.*

4 Starting Mars

To run Mars, run `mars.jar` (for Java 1.4 or later) or `mars-j13.jar` (for Java 1.3), either by double-clicking on it (in the Macintosh Finder or Windows Explorer) or by changing to the Mars home directory and typing `java -jar mars.jar`.

When running Mars from the command line, you can specify a previously-saved Mars configuration file to open on startup. In addition, the following command-line options are available:

- home** specify a “home” directory from which to load extensions and the probe definitions file. Defaults to `.` (the current working directory).
- nogui** start without the graphical user interface. This is useful for running the mail notification plugin or seeing status changes on the console. Requires a configuration file to be given.
- geom** start the graphical user interface with the specified geometry, expressed as width and height in pixels joined with an 'x', for example, '800x600'.

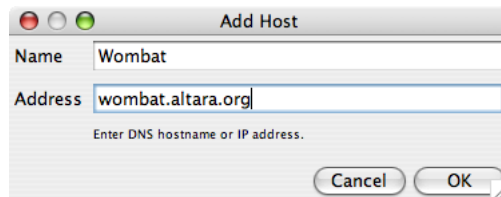


Figure 2: The Mars host editor

Example: to run Mars in `/usr/mars/mars.jar`, loading plugins from `./mars/` and the configuration file `/mars-config.xml` without starting the user interface, type:

```
java -jar /usr/mars/mars.jar --nogui --home ./mars
/mars-config.xml
```

When Mars starts, the splash screen will appear momentarily, informing you of any extensions it finds and loads on startup. Then the main window appears (see figure 1). The Mars main window has three parts: a host/service tree, a fault list, and a service detail panel. The host/service tree displays the hosts and services Mars knows about; when Mars is actively monitoring, it also displays the status of each service. The fault list displays all current problems with monitored services. The service detail display shows relevant details about the service selected in the host/service tree.

5 Adding Hosts and Services

The host/service tree can be manipulated in two ways: through a context menu on the tree, or via keyboard commands.

By right-clicking (control-clicking on Macintosh) on a host or service in the tree, or on blank space within the tree, you can bring up the context menu, which allows you to access relevant tree editing operations. Since you've started with a new configuration, the host/service tree is empty; anywhere you right-click will bring up a context menu with a single option, `New Host...`. Select this option to bring up a host editor on a new host.

5.1 The Host Editor

The host editor (see figure 2) has two fields:

Name is the name by which you want Mars to refer to the host; it could be the host name without the domain, or a descriptive name (e.g. "Web Server") completely unrelated to the host's actual DNS name.

Address is the fully qualified DNS name or IP address of the host. Mars will perform a lookup of the given name before adding the host to the

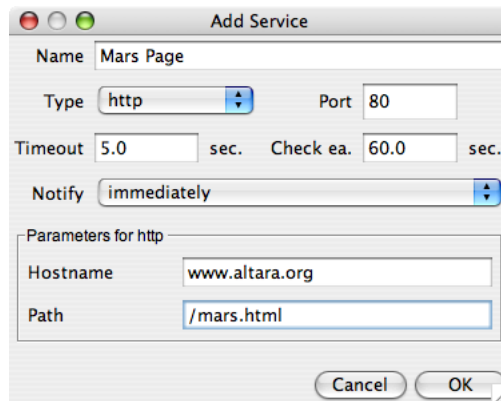


Figure 3: The Mars service editor

host/service tree, and will not allow you to continue unless the given name resolves to an address; this ensures you don't get false alarms about services being down because of a typo in the `Address` field.

5.2 The Service Editor

Now that you've defined a host, you need to define a set of services to monitor on that host. Right-click (control-click) on the host name to bring up the host context menu and select `Add Service...`, which will bring up a service editor on a new service on the selected host.

The service editor (see figure 3) has the following fields:

Name is, as for a Host, simply a label by which Mars will display the service. Suggestions are the name of the service ("http"), the server software ("Tomcat"), or the user-visible application the service provides ("PR System").

Type is the type of service – Mars uses this to know how to probe a service. Currently, there are seven available service types (more services may be added by installing additional probe extensions):

http probes HTTP/1.1 servers. It is useful for checking any web service. Through its `Hostname` parameter, you can specify the HTTP/1.1 `Host:` request header to send with each request. This can be used to differentiate named virtual hosts during probing. The default hostname is the address of the host on which the service is running. Through its `Path` parameter, you can specify any path on the given server to monitor. The `Path` is the portion of the URL after the host and port, starting with the first slash. The default path is `/`.

http-regex probes HTTP/1.1 servers, but retrieves the body of the document along with the header, then tests the body against specified regular expressions for success and failure. If the failure expression matches in the body, the service will fail with status **bad reply**. If neither the success nor the failure expressions match in the body, the service will fail with status **closing**. Expressions left blank will not be matched.

ftp probes FTP servers.

smtp probes SMTP (Mail Transfer Agent) servers.

imap probes IMAPv4 servers.

pop3 probes POP3 servers.

ssh probes Secure Shell servers (either version 1 or 2).

tcp-connect probes any TCP-based service for connection. **tcp-connect** *does not test any protocol interaction; it only ensures that the host is accepting connections on the given port.* Use **tcp-connect** to monitor services whose protocols Mars does not yet support, but understand that this service type is less reliable at detecting failure than the others.

Port is the TCP port on the host on which the service is running. Each time you select a new **Type**, this changes to the default port for the given service type.

Timeout is the time, in seconds, Mars will wait for a service to respond before considering it timed out. The default is 5.0 seconds, but this may be far too long or far too short, depending on the service and your environment. If you notice a service that is up reporting back as timed out very often, consider increasing this value.

Check ea. (n) sec. is the minimum time Mars will delay between running probes of a service. Use longer delays here to decrease monitoring network traffic.

Notify determines how Mars will treat status change notifications for this service. Note that setting the notification behavior here *will not* have any effect unless you also enable at least one notification plugin (see section 7). The following notification behaviors are available:

never causes Mars never to send notifications for this service. Use this for services whose status is not important enough for notification.

immediately causes Mars to send notifications immediately on any service status change. This is the default behavior, and the behavior of Mars versions prior to 2.2.2.

on second attempt causes Mars to wait for two consecutive probes to return the new status before notifying of the status change. Use this

to prevent transient failures (e.g. timeout due to temporary network congestion) from triggering notifications.

on third attempt similarly waits for three consecutive probes. Use this to be even more aggressive about filtering out transient failures.

Parameters for... is a panel in which service-type specific parameters are set. Currently, only **http** and **http-regex** use this panel.

Repeat this process until all the hosts and services you wish to monitor with Mars have been added to the host/service tree. If you have multiple hosts that have the same services, you can use the **Duplicate Host...** context menu item to create a new host with the same services as an existing host.

Once you've finished configuring Mars, you'll probably want to save your configuration; do this by selecting **Save** from the File menu, clicking the save button on the toolbar, or typing **Ctrl-S** (**Cmd-S** on Macintosh).

This will bring up a dialog asking you for the name of the file to save the configuration into; the configuration will be written as an XML file, the format of which is described in the *Hacker's Guide*.

5.3 Keyboard Editing

The host/service tree can also be edited entirely with keyboard commands, as follows:

up/down arrows Select next or previous host or service.

left arrow Expand selected host, select first service.

enter Edit the selected host or service.

delete Delete the selected host or service.

Ctrl-T/Cmd-T Create a new host.

Ctrl-R/Cmd-R Create a new service on the selected host.

Ctrl-D/Cmd-D Duplicate the selected host.

6 Monitoring Services with Mars

To begin monitoring services with Mars, select **Start** from the **Control** menu, click the start button in the toolbar, or type **Ctrl-B** (**Cmd-B** on Macintosh). Mars will begin monitoring the services you have specified, display each service's status in the host/service tree, and display any faults in the fault list (see figure 4). While Mars is actively monitoring, the number of faulted services appears in the title bar. This allows you to minimize the Mars window and still keep a summary eye on your network.

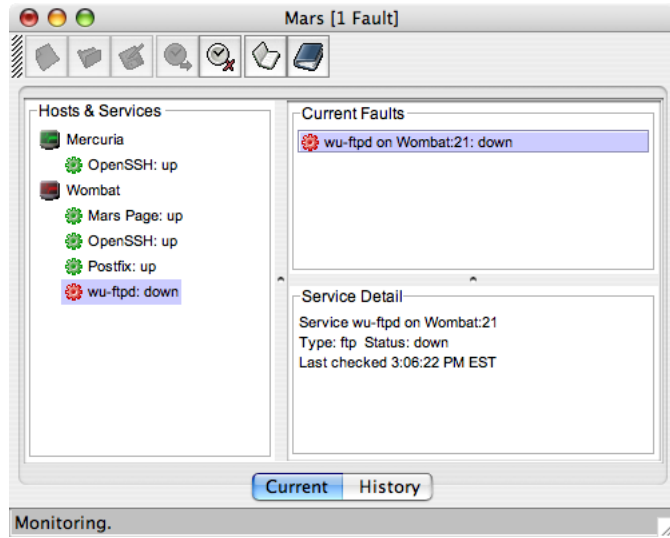


Figure 4: Mars in action

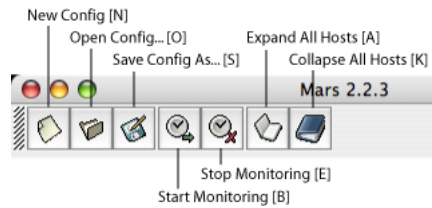


Figure 5: Mars toolbar

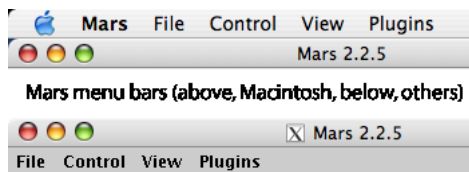


Figure 6: Mars menubars

To see a host's services in the host/service tree, double-click on the host's name or icon. To see detailed information for a service, click on that service to select it; the detailed information will be displayed in the service details pane. You can also click on a faulted service in the fault list to see that service's details.

To fully expand the host/service tree, select **Expand Hosts** from the **View** menu, click the expand button in the toolbar, or type **Ctrl-A** (**Cmd-A** on Macintosh). To collapse the tree (and view only hosts), select **Collapse Hosts** from the **View** menu, click the collapse button in the toolbar, or type **Ctrl-K** (**Cmd-K** on Macintosh).

A service's status is expressed as one of the following:

down The host is not accepting connections at the specified port. This generally means the service is not running, or the host is not connected to the network. This fault can also occur if there is a firewall between the host and the machine on which Mars is running set to deny connections to the specified port. Check your firewall configuration to ensure the host is actually down.

closing The host is accepting connections at the specified port, but is closing the connection unexpectedly during the probe. This status is returned for **http-regex** services when neither the success nor failure expressions match. Another common cause of this is services wrapped with **tcpd**, or other IP-address based access control software, set to deny connections from the machine on which Mars is running. Check your wrappers configuration; if connections should be allowed from the machine on which Mars is running, there is something wrong with the service.

bad reply The service is returning unexpected data during the probe. The service may or may not be down; check the "Received" line in the service details pane to see what the service returned. For **http** services, this means the server returned an error result code (4xx or 5xx).

timeout The service is not giving an expected response during the specified timeout period. This may indicate excessive load on the service. If you see this fault often, and the service is operating properly, consider increasing the timeout delay for the service.

up The service is operating properly, and responding within the specified timeout.

probe error An unspecified Mars error occurred during the probe of the service. This may occur during shutdown, or as the result of a bug in Mars itself.

unknown The service has not been probed since Mars started, or since it was first added. Wait a moment for the probe to finish running.

If a host has no faulted services, its icon appears green in the host/service tree while Mars is monitoring. If any of a host's services are faulted, that host's icon will appear red.

If you'd like to ignore a fault but keep monitoring the service, you may hide that fault by right-clicking in the fault list and selecting **Hide** from the menu that appears. The service is still being monitored; it's just hidden in the fault list. Hidden faults may be shown via the **Show All** menu item.

The history of each service's status is displayed in the recent changes list. The recent changes list is accessible via the **History** tab in the Mars window. This list shows the 200 most recent service status changes, and can be cleared using the **Clear History** button.

To stop Mars from monitoring, select **Stop** from the **Control** menu, click the stop button in the toolbar, or type **Ctrl-E** (**Cmd-E** on Macintosh). You'll notice when Mars is stopped, all current faults disappear from the fault list, and the host/service tree reverts to showing no status for each service.

7 Plugins and Probe Extensions

Mars supports plugins, independent modules of Java code that can listen for service status changes and act upon them accordingly. Plugins are dynamically loaded from the Mars home directory at startup time, and are listed and configured via the **Plugins** menu. Plugins are shipped in the **extras** directory in the Mars distribution, and will need to be moved to the home directory in order to load at startup time.

Mars also supports probe extensions, dynamically loaded Java classes which add new service monitoring capabilities to Mars. Probe extensions are shipped in the **extras** directory within the Mars distribution; you'll need to move them to the Mars home directory in order for them to load at startup time. Each of these probe extensions adds a service type to the available services Mars can probe.

7.1 UI Notification

The UI Notification plugin displays a dialog box when certain service status changes occur. To install UI Notification, move `plugin.swingnotify.jar` from the **extras** directory into your Mars home directory.

To enable UI notification, select it in the **Plugins** menu; the UI notification editor will pop up. Select the **Enabled** checkbox to enable notification. Select **Beep on fault** to cause Mars to beep when displaying a notification. The **Notify when a service comes back up** checkbox causes Mars to display another dialog when a service comes back up.

UI Notification requires the Swing user interface, and will be disabled if the `--nogui` command-line option is in effect.

7.2 Mail Notification

This plugin uses a very simple SMTP client to send a very simple message to a given list of e-mail addresses using a single given mail server whenever a service is detected as faulted. To install Mail Notification, move `plugin_mailnotify.jar` from the `extras` directory into your Mars home directory.

To enable Mail Notification, select it in the `Plugins` menu; the mail notification editor will pop up. Select the `Enabled` checkbox to enable notification, then fill in the `SMTP Server` field with the address or DNS hostname of a SMTP server configured to relay mail from the machine on which Mars is running.

To specify a set of recipients for the notification e-mail, enter the addresses into the `Recipient(s)` field separated by commas, as you would in most popular mail client programs. To specify a “From” address to appear on each notification e-mail, place the address in the `Sender` field. If you leave this field blank, it will default to the first address in the recipient list. Note that the addresses in these fields must be of the form `host@subdomain.domain.dom` only; you must omit angle brackets, or proper names, quoted or not.

As with the UI Notification plugin, the `Notify when a service comes back up` checkbox causes Mars to send another message when a service comes back up.

With the Mail Notification plugin enabled, you can run Mars “headless” – i.e., without a GUI. To do this, first run Mars to create a configuration file with the hosts and services you wish to monitor, and with Mail Notification configured and enabled. Save this config file, then invoke Mars from the command line with the `--nogui` option. Mars will print status messages to the console and send email notifications when services go down or time out.

7.3 CSV Logger

This plugin logs the results of each probe to a CSV (comma-separated values) file, in the following format: `date, time, host-name, service-name, host-address, port, service-type, status, response-time`. The date and time are given in ISO-8601 format (`year/mo/dy hr:mn:sc.msc`) Response time is given in milliseconds. To install CSV logging, move `plugin_csvlog.jar` from the `extras` directory into your Mars home directory.

To enable CSV logging, select it in the `Plugins` menu; the CSV logging editor will pop up. Select the `Enabled` checkbox to enable logging, then fill in a pathname to a CSV file to log to in the `Log file` field. You can use the `Choose...` button to select a pathname using the file browser, if you like. The selected log file path must be within an existing directory and writable by you.

If the given log file exists, the CSV Logger plugin will append new CSV log entries to the end of it. This means you can log probe information across multiple runs of Mars without having to reconfigure the CSV Logger or lose the old data each time.

With the CSV Logger plugin enabled, you can run Mars “headless” – i.e., without a GUI. To do this, first run Mars to create a configuration file with the hosts and services you wish to monitor, and with the logger configured and

enabled. Save this config file, then invoke Mars from the command line with the `--nogui` option. Mars will print status messages to the console and write a log to the specified logfile.

7.4 XML Snapshot

This plugin periodically dumps status information on the services Mars is monitoring into a named XML file. This XML file has the same format as the Mars configuration file (see the Hacker's Guide for more), but with status information in the `status` element within each `service`. This plugin is designed to be used with XSL transformations to display status information from Mars via a web interface. To install the XML Snapshot plugin, move `plugin_xmlsnap.jar` from the `extras` directory into your Mars home directory.

To enable the XML Snapshot plugin, select it in the `Plugins` menu; the XML Snapshot editor will pop up. Select the `Enabled` checkbox to enable logging, then fill in a pathname to an XML file to dump to in the `XML file` field. You can use the `Choose...` button to select a pathname using the file browser, if you like. The selected path must be within an existing directory and writable by you.

XML snapshots are written every 60 seconds by default; to change this, enter a new time in the `Period` field.

With the XML Snapshot plugin enabled, you can run Mars “headless” – i.e., without a GUI. To do this, first run Mars to create a configuration file with the hosts and services you wish to monitor, and with the logger configured and enabled. Save this config file, then invoke Mars from the command line with the `--nogui` option. Mars will print status messages to the console and dump status information to the specified snapshot file.

7.5 XMPP Notification

This plugin sends notification via an XMPP (Jabber) instant-messaging server whenever a service is detected as faulted. The XMPP Notification plugin includes the Smack XMPP client library version 1.2.1, ©2002-2003 Jive Software, Inc., all rights reserved. Smack is developed by Jive Software, and is available at <http://www.jivesoftware.com/xmpp/smack>. To install XMPP notification, move `plugin_xmpp.jar` from the `extras` directory into your Mars home directory.

To enable XMPP notification, select it in the `Plugins` menu; the XMPP notification editor will pop up. Select the `Enabled` checkbox to enable notification, then fill in the form. `XMPP Server` and `XMPP Port` identify the server to connect to use to send notification messages. `Username` and `Password` are the credentials used to connect to the server. The string in `Prefix` will be appended to each message sent by the plugin. `Recipient(s)` is a list of Jabber addresses to send the notification to. Separate multiple addresses with commas.

With the XMPP Notification plugin enabled, you can run Mars “headless” – i.e., without a GUI. To do this, first run Mars to create a configuration file with

the hosts and services you wish to monitor, and with the plugin configured and enabled. Save this config file, then invoke Mars from the command line with the `--nogui` option. Mars will print status messages to the console and send notifications via XMPP.

7.6 Client Debugger

This plugin provides a detailed view of Mars' interaction with remote hosts in real time. It is intended for debugging XML-defined probes in `mars-def.xml` and probe extensions. To install the client debugger, move `plugin_cdebug.jar` from the `extras` directory into your Mars home directory.

To use the client debugger, select it in the Plugins menu; the client debugger preferences will pop up. Select the **Enabled** checkbox and set the Horizon slider to the number of sessions you want the debugger to remember.

A new tab, **Debug**, will appear in the Mars window. This tab is split into two panels, a debug session list and a session detail display. Debug sessions are displayed by timestamp (most recent first), the name of the client interface producing them and some identifying information about the connection. XML-defined probes (and the Mars mail notification client) are implemented by a class called `SendExpectClient`, so these sessions are defined by the client interface `SEC`. The included `HTTPS` and `JDBC` probe extensions are identified, respectively, as `HTTPS` and `JDBC`.

By selecting a session in the upper panel, the session's details will appear in the lower panel. There are three types of entries in a session: data sent (green right double arrow), data received (yellow left double arrow), and information ("i" in a blue circle). Sent and received data is split into lines; the special symbols `<CR>` and `<LF>` are used for carriage return and linefeed. The timestamp on each entry is the time of the sending/receipt of the first character in the entry.

The Client Debugger requires the Swing user interface, and will be disabled if the `--nogui` command-line option is in effect.

7.7 Mac OS X Integration

This plugin provides integration between Mars and the Mac OS X environment. It is primarily intended for use by the `Mars.app` Mac OS X application-bundle version of Mars, and is not included with the non-Macintosh binary distribution. See the section 8 for more.

7.8 HTTPS Probe Extension

This probe extension adds a new service type, `https`, to the available service types, which probes secure HTTP servers. To install the HTTPS probe extension, move `probe_https.jar` from the `extras` directory into your Mars home directory.

The `https` service takes three parameters. `Hostname` and `Path` are used as they are with the other http services. `Content`, if present, is a plain text string

to scan for in the response body. It is *not* the same as the regular expressions supported by the `http-regexp`

Note that the `https` service appears to report failure for any server presenting a certificate not signed by a certificate authority in the `lib/security/cacerts` file in your Java installation. Use the `keytool` program included with Java to modify this file.

7.9 JDBC Probe Extension

This probe extension adds a new service type, `jdbc`, to the available service types, which probes JDBC-accessible database servers. To install the JDBC probe extension, move `probe_jdbc.jar` from the `extras` directory into your Mars home directory.

The `jdbc` service takes five parameters. `Protocol` is the database protocol to use (the protocol part after `jdbc:` in a JDBC URL). `Driver` is the name of the Java class of the JDBC driver to use. This driver must be in the system classpath at Mars startup time to be usable by the JDBC probe. `[Database]` is the name of the database to connect to, and `[Username]` and `[Password]` are the credentials to present.

8 Mars on Mac OS X

A version of Mars built to take advantage of Mac OS X's support for Java applications is available separately as a `.dmg` disk image file. This disk image contains a Mac OS X installer package which will place a Mac-friendly Mars in your `Applications` directory. This manual applies in full to the Mac OS X version of Mars with a few caveats:

- Mars for Mac OS X requires Java 1.4. You can run the non-Macintosh version of Mars on Java 1.3, but it won't integrate nicely with Mac OS X.
- Mars can be started simply by double-clicking the application in `Applications`.
- The Mars home directory is contained in `/Library/Leapfrog/Mars`. To install Mars extensions, you'll need to place them in that directory. All included extensions are already installed by the Mars installer package.
- To open configuration files from the command line, use `open -a Mars`.

The Mac-friendly version of Mars provides several benefits, as follows:

- The Mars menu bar appears at the top of the screen, as with other Macintosh applications, and not in the window as with the non-Macintosh version.
- The Mars window's close box ("jellybean?") will display the "dirty" dot if the configuration has been changed since the last save.

- The Quit item on the application menu works properly; with the non-Macintosh version, this menu item will always quit without asking whether any unsaved changes should be saved.
- The About Mars... item on the Application menu will re-display the Mars splash screen.
- Mac-friendly Mars has a pretty (or at least identifiable) dock icon.
- Mars' dock icon will accept drag-and-drop of XML files for use as configuration files.

A Change History

A.1 Mars 2.2.7 – March 22, 2004

Mars 2.2.7 is a bugfix release. The following changes were made:

- SimpleSmtpClient (used in the mail notification plugin) now properly uses CR/LF to terminate lines sent to the mail server. This, like the bug fixed in 2.2.3 in the probe engine, was not technically RFC-compliant, and was causing problems with mail notification on Microsoft servers. Thanks to Sebastian Johnck and Aaron Roller for their help with this bug.
- Bugfixes to JDBC Probe and tcp-connect service types, thanks to S. Çağlar Onur.
- XMPP notification now properly handles multiple recipients, thanks to a fix submitted by Paul Goulbourn.

A.2 Mars 2.2.6 – March 1, 2004

Mars 2.2.6 is a feature enhancement and bugfix release. The following changes were made:

- Plugins may now display their own tabs in the Mars main window. This functionality is intended for the construction of data visualization plugins, and is used by the new Mars Client Debugger.
- A new Client Debugger plugin has been added. This allows users to view all Mars' interactions with the network, in real time, and is primarily intended to aid in the development of new probe extensions and XML-based probes.
- All included probe extensions and plugins have been instrumented to report to the client debugger.
- All included probe extensions and plugins are now distributed in the `extras` directory.

- Timestamps have been added to console status messages in `--nogui` mode.
- The service detail view now displays the next scheduled probe time.
- Plugins that use the user interface will now refuse to load in `--nogui` mode.
- A new XML Snapshot plugin has been added. Thanks to Robert Fuller for working on this.
- A minor bug which occasionally caused the `received` property of XML-defined probes to be corrupted with an internal end-of-stream marker has been fixed.
- Mars now explicitly uses UTF8 encoding for its configuration file.

A.3 Mars 2.2.5 – February 10, 2004

Mars 2.2.5 is a bugfix and feature enhancement release. The following changes were made:

- Loading configuration files from the command line works again. This was, officially, a Really Embarrassing Bug.
- XMPP notification bugfixes. XMPP client connection is now properly stopped on disable, and restarted on configuration change. Thanks to Mike Dixon for his help in tracking this down.
- Tweaked open/save dialogs to actually change to the correct directory on all platforms. Default to saving new configuration files in home directory. Fixed minor bugs in configuration file handling.
- Mars' user interface has undergone a minor overhaul. Mars now has a menu bar. The host/service tree is editable using the keyboard. The Config panel has been replaced by the Plugins menu.
- Mac OS X integration is Mac-friendlier. Mars now uses the system menubar and the “dirty” dot in the close jellybean. Configuration file open through the Finder now works properly even if Mars wasn't running yet. Oops.

A.4 Mars 2.2.4 – February 4, 2004

Mars 2.2.4 is a feature enhancement and bugfix release. The following changes were made:

- JDBC is now supported by an included probe extension, thanks to Jon Steele.
- Notification via XMPP (Jabber) is now supported by an included plugin, thanks to Robert Fuller.

- Mars now integrates with Mac OS X (Java 1.4 or greater) via an included plugin. Mars is now additionally distributed as a “native” Mac OS X application.
- Tweaked client code to put packets on the wire only when a send was completed; previously, Mars would send many unnecessarily small TCP segments when probing, especially HTTP and its ilk. While this behavior was not technically incorrect, it could theoretically cause problems with some poorly coded servers, and was (ever so slightly) disrespectful of the network.

A.5 Mars 2.2.3 – January 26, 2004

Mars 2.2.3 is a feature enhancement and bugfix release. The following requests have been addressed:

611897 HTTPS is now supported by an included probe extension, thanks to Scott Ahten.

726603 Added expand and collapse toolbar buttons, for quicker access to the host/service tree, at the suggestion of Richard Gill.

838540 Fixed bug that would cause Mars to send LF where CRLF is required. Apparently, most (Unix-based) services don't really care about the CR. This should fix up problems on probing certain Windows-based services. Thank you, Noel Bergman.

851943 Changed all date output to conform to ISO 8601.

851944 Added ability to reference a regular expression contained in a service parameter, and the http-regexp service to take advantage of this. Thanks to Robert Fuller.

851945 Added the ability to specify the geometry of the Mars window on the command line, at the suggestion of Stefan Svensson.

851946 Added code to extension loader to allow multiple plugins to reside in the same .jar file. This is not presently used by the shipping Mars system, but could be used by plugin/probe developers to avoid clutter in the Mars home directory. Thank you, Ivica Loncar.

A.6 Mars 2.2.2 – March 11, 2003

Mars 2.2.2 is a feature enhancement and bugfix release. The following requests have been addressed:

590319 Added host icon coloring according to service status.

590333 Added CSV Logger plugin

- 631869** Added ability to choose notification retry count by service. Added `NotificationListener` interface to plugin framework and `Notifier` class to engine to support delayed notification.
- 683792** Fixed mail notification bug that caused problems with some SMTP servers which require MAIL and RCPT addresses to be enclosed in angle brackets. Thanks to Laurent Marot for helping me track this one down.
- 692534** Added ability to send notifications to multiple e-mail recipients.
- 693452** Fixed bug that would cause the information in the status bar to fall behind the monitoring engine when Mars is very busy.
- 693456** Added `Hostname` to `Monitor` parameter to HTTP probe. Added `%(remote-hostname)` parameter hack to `mars-def.xml` to support this.
- 700452** Updated Jakarta ORO and Xerces-J packages.

A.7 Mars 2.2.1 – February 25, 2003

Mars 2.2.1 is a feature enhancement and bugfix release. The following requests have been addressed:

- 590325** Added a status bar.
- 590328** Added dynamic loading of plugins and probe extensions.
- 632639** The UI Notification plugin now runs in a separate thread, to keep from blocking other status change notifications. Thanks to Mark Lewis for sending in this patch.
- 629689** The Mail Notification plugin now allows mail to be sent from a different address than the recipient address. Thanks to Mark Lewis for sending in this patch.

A.8 Mars 2.2.0 – October 28, 2002

Mars 2.2.0 is an under-the-hood feature enhancement and bugfix release. The following requests have been addressed:

- 594660** Added the ability to define probes using XML.
- 602762** Fixed a bug in SMTP probing which was looking for an incorrect response from SMTP servers.

A.9 Mars 2.1.3 – August 12, 2002

Mars 2.1.3 is a feature enhancement release. The following features have been added:

- 590315** Added the ability to hide faults in the fault list.
- 590316** Added the ability to clear the status history.
- 590323** Added the Duplicate Host... command to the service tree context menu, which makes a copy of a given host with a different name and address.
- 590330** Added the ability to send a notification when a service comes back up to the mail notification plugin.
- 593098** Bound the selection in the fault list to the service tree, so that when the user clicks on a fault in the fault list, that service is selected in the service tree (which also causes it to be displayed in the detail display).

A.10 Mars 2.1.2 – July 14, 2002

Mars 2.1.2 is a bugfix/feature enhancement release. Fixed in this release is the bug/misfeature that forced the user to stop monitoring before editing the host/service tree - hosts and services can now be added, edited, or deleted at any time. Also added is the ability to specify a path to an HTTP probe; now you can use Mars to check to make sure your web applications and services are running properly (i.e. not replying 500 Server Error, which will show up as a “bad reply” in Mars) - not just to check that your web server itself is running.

A.11 Mars 2.1.1 – June 30, 2002

Mars 2.1.1 is a bugfix release. Fixed in this release are a bug in the probe client that caused all sorts of problems on Win2k (namely a CPU usage spike and an enormous memory leak), and a minor design flaw in the HTTP probe. Formerly, an HTTP server had to respond to the probe with a 200 (OK) response status to be considered “up”; valid “up” responses such as the 3xx redirect codes were shown as “timed out”. No new features were added in this release.

A.12 Mars 2.1.0 – June 5, 2002

Mars 2.1.0 is a bugfix/feature enhancement release. The user interface and functionality is largely the same. In addition to various user interface tweaks and bug fixes, Mars 2.1.0 features:

- A rewritten probe client backend that fixes a couple of rather embarrassing bugs that can cause Mars to hang. This rewritten backend properly handles services that accept connections then fail to respond without hanging Mars. It’s also more flexible, and may be used in the future to allow simple TCP/IP text based probes to be described in XML.

- A properly multithreaded probing engine. Though Mars 2.0.0 was designed to support multiple probe threads running at once, only one probe thread was used for simplicity. The probe engine now properly starts the correct number of threads to ensure that “probe each n sec.” parameters on services are met, within reason.
- Support for a plugin framework, though planned autoload of plugins from .jar files has not yet been written (autoload is supported in Mars 2.2.1). A plugin for notifying of service faults via email was added, as well.

A.13 Mars 2.0.0 – May 8, 2002

Mars 2 is a complete rewrite of Mars. The first and most noticeable change is that the user interface is completely different. The second is that the configuration file format is completely different. In addition, several Mars 1.x features have been removed, including:

- SPOTS. Mars can no longer talk to the SPOTS agent, which provides load average, memory usage, and filesystem usage information on Unix systems. SPOTS is inconsistent with a strictly services-oriented view of a host, though I may add some sort of agent support like SPOTS in future versions if enough of you really miss it.
- Notification of down services, via the GUI and email. (Notification is supported in Mars 2.1.0.)