

Appmon

version 2.0

Typeset in \LaTeX from SGML source using the DOCBUILDER 3.0 Document System.

Chapter 1

Appmon User's Guide

The Application Monitor, *Appmon*, is a graphical utility used to supervise applications executing either locally or on remote nodes. The process tree of an application can furthermore be monitored.

1.1 Appmon

Introduction

The application monitor Appmon is a graphical node and application viewer. The tool shows an overview of all applications on all known nodes, and it is possible to view the process tree for an application running on any of the nodes.

Getting Started with Appmon

Start Appmon by calling `appmon:start()`. It will start the main window [page 2] showing a load meter and the applications running at the current node. By clicking on one of the applications a window showing the process tree of the application will be opened, the application window [page 4].

The main window is equipped with a menubar from which it is possible to:

- exit Appmon
- perform some operations on the node
- set how information should be displayed
- select which node to show
- open help (this document).

The application window shows the process tree for an application with each process displayed as a box. It is possible to view information about the processes, to send messages to them, and to trace and kill them.

The Main Window

The main window shows a load meter and all applications running at the displayed node. Select which node to display in the window by choosing the node name from the Nodes menu. It is also possible to run Appmon in a many-window mode where a new instance of the main window is opened for each node to be displayed.

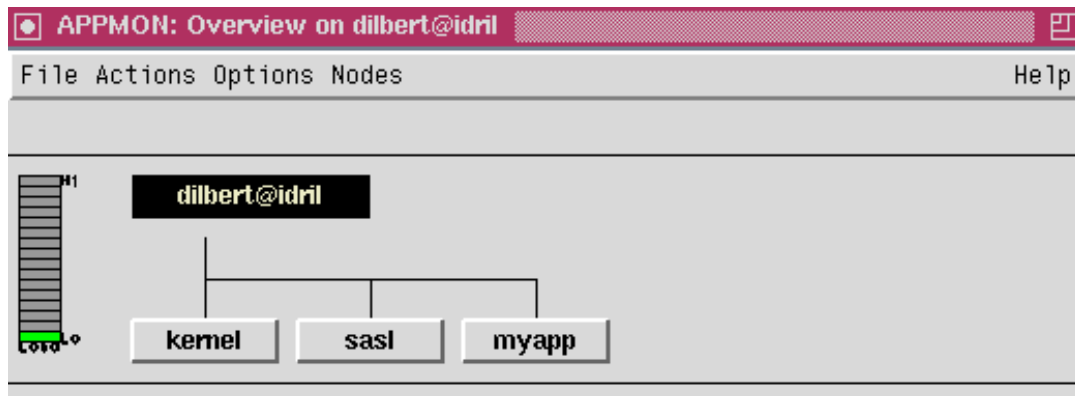


Figure 1.1: The Main Window.

The load meter shows load measured as processor time, or as the length of the ready queue.

Every application running at the node is shown as a button. Clicking the button will open the application window showing the process tree for the application.

The File Menu

Show List Box... This will open the listbox window [page 6] which lists all nodes and applications.

This window can be more easy to use than the normal, graphical user interface when the system consists of a large number of nodes and/or applications.

Close Close the window. If no other instance of the main window exists, Appmon will be stopped.

Exit Stop Appmon.

The Actions Menu

Reboot Call `init:reboot()` at the currently displayed node. This will stop the node. If the `-heart` system flag was given, the heart program will try to reboot the system.

Restart Call `init:restart()` at the currently displayed node. This will restart the node.

Stop Call `init:stop()` at the currently displayed node. This will stop the node.

Ping Call `net:ping(Node)` where `Node` is the currently displayed node. This can be useful when the connection to the node has been lost.

The Options Menu

One window/Many windows Select one of these radio button to run Appmon in one-window or many-window mode. In many-windows mode, a new instance of the main window is opened for each node to be displayed. The default value is one-window mode.

Load: time/queue Select one of these radio buttons to either calculate load as processor time or as the length of the ready queue, which is the number of processes ready to execute. By default, the load is calculated as processor time.

Load: progressive/linear Select one of these radio buttons to show load either according to a progressive or a linear scale. By default, the load is shown according to a progressive scale.

The Nodes Menu

The Nodes menu contains all currently and previously known nodes. Selecting a node from the Nodes menu will cause that node to be displayed in either the same window, or in a new window, depending on if Appmon is run in one-window or many-windows mode. If the connection to the node has been lost, this will be shown in the window.

The Help Menu

Help Selecting Help from the Help menu will cause the HTML version of the Appmon User's Guide (this document) to be displayed. Currently this function requires Netscape to be up and running.

The Application Window

The application window shows the process tree for an application. The window title contains the application name and the node name. The window also contains a menubar and a toolbar.

The application window can be opened from the main window by clicking on the button denoting the application, or from the listbox window by selecting the application and clicking on the *Load* button.



Figure 1.2: The Application Window.

The application can be shown either as a strict supervision tree, or as a process view with all linked processes. In supervision mode, the tree-gathering and -building algorithm assumes conformance to the OTP design principles.

The File Menu

Close Close the application window.

The Options Menu

Refresh Refresh the application window.

Sup. view/ Proc. view Select one of these radio buttons to show the application as a strict supervision tree, or as a process view with all linked processes. By default, the process view is used.

The Toolbar

The toolbar consists of four buttons: *Info*, *Send*, *Trace* and *Kill*. First select one of these buttons and then select to which process the action should apply by clicking on a process in the process tree. By default *Info* is selected which means that clicking on a process, without selecting *Send*, *Trace* or *Kill* first, will open the process information window [page 7].

Info Open the process information window [page 7], which displays the information about the process given by `process_info(Pid)`.

Send Send a message to a process. A window is opened where the message can be specified. Click *Ok* to send the message or *Cancel* to cancel.

Trace Switch on OTP tracing of a process. `sys:trace(Pid,true)` is called. Selecting *Trace* a second time for the same process will switch the tracing off instead.

Kill Kill a process. `exit(Pid,kill)` is called.

The Listbox Window

The listbox window lists all nodes and applications. This window can be more easy to use than the normal, graphical user interface when the system consists of a large number of nodes and/or applications.

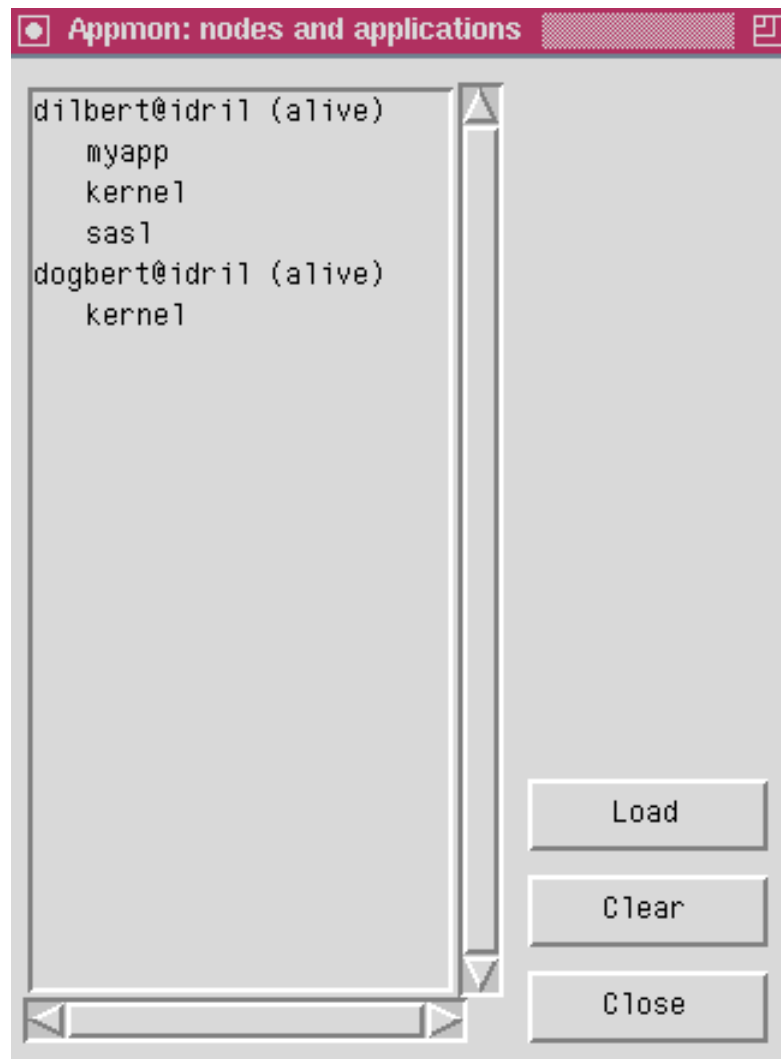


Figure 1.3: The Listbox Window.

The window contains three buttons:

Load First selecting an application and then clicking the *Load* button will open the application window [page 4] for that application.

Clear De-select any selected node or application name.

Close Close the listbox window.

The Process Information Window

The process information window shows information about different processes as given by `process_info(Pid)`.

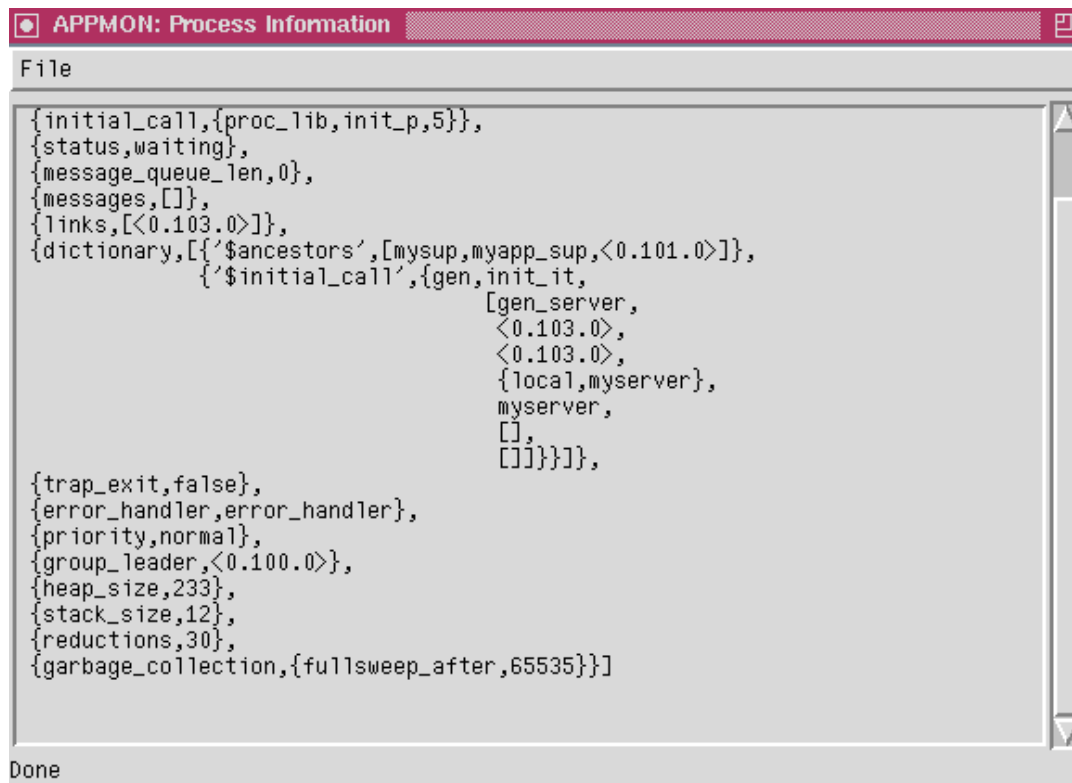


Figure 1.4: The Process Information Window.

The File Menu

Close Close the process information window.

Appmon Reference Manual

Short Summaries

- Erlang Module **appmon** [page 10] – A graphical node and application process tree viewer.

appmon

The following functions are exported:

- `start()`
[page 10] `start/0` starts Appmon.

appmon (Module)

The application monitor Appmon is a graphical utility used to supervise applications executing either locally or on remote nodes. The process tree of an application can furthermore be monitored.

Exports

`start()`

`start/0` starts Appmon.

See Also

[Appmon User's Guide](#)

List of Figures

Chapter 1: Appmon User's Guide

1.1	The Main Window.	3
1.2	The Application Window.	5
1.3	The Listbox Window.	7
1.4	The Process Information Window.	8

Index

Modules are typed in *this* way.
Functions are typed in *this* way.

appmon
start/0, 10

start/0
appmon , 10