

Oz Panel

Christian Schulte

Version 1.3.2
June 15, 2006



Abstract

The Oz Panel is a graphical tool to visualize and control the most important parameters of a running Mozart system.

Credits

Mozart logo by Christian Lindig

License Agreement

This software and its documentation are copyrighted by the German Research Center for Artificial Intelligence (DFKI), the Swedish Institute of Computer Science (SICS), and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE AND ITS DOCUMENTATION ARE PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Contents

1	Usage	1
1.1	Information on Threads	2
1.2	Information on Memory	3
1.3	Information on Problem Solving	3
1.4	Information on Programming Interface	4
2	The Menu Bar	5
2.1	The <code>Panel</code> Menu	5
2.2	The <code>Options</code> Menu	5
3	The <code>Panel</code> Object	7
3.1	Invoking	7
3.2	Options	7

Usage

This chapter briefly outlines the features of the Oz Panel.

invoking The Panel is invoked by either feeding the expression

```
{Panel.object open}
```

or by selecting from the Oz Menu in the Oz Programming Interface the `Panel` entry. Furthermore, it is possible to invoke the Panel from the Oz Programming Interface with the key C-C p.

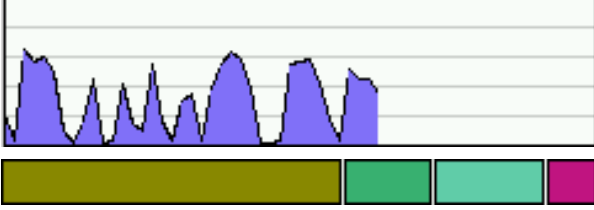
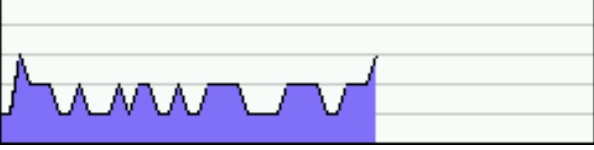
monitor The Panel monitors the most important system parameters. The Panel can be configured (see Section 2.2) to also allow to

configure configure system parameters. The Panel works as a graphical frontend for the procedures `Property.get` and `Property.put` that are described in Chapter *Emulator Properties: Property, (System Modules)*.

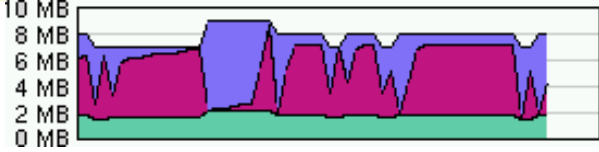
update The panel periodically updates its display. By default, only when the mouse pointer is over the Panel's window updating takes place. However, this can be changed (see Section 2.2).

The information the panel displays is described in Section 1.1, Section 1.2, Section 1.3, and Section 1.4.

1.1 Information on Threads

Runtime Field	Explanation
Run	Run time.
Garbage collection	Time spent on garbage collection.
Copy	Time spent on copying (that is, on cloning of spaces).
Propagation	Time spent on executing propagators.
	<p>Operating system load of the Mozart engine.</p> <p>Relative distribution of entire computation time.</p>
Threads Field	Explanation
Created	Total number of threads created.
Runnable	Number of currently runnable threads.
	Number of currently runnable threads..
Priorities Field	Explanation
High / Medium	Relation between time slices available for threads of priorities <code>high</code> and <code>medium</code> .
Medium / Low	Relation between time slices available for threads of priorities <code>medium</code> and <code>low</code> .

1.2 Information on Memory

Heap Usage	
Field	Explanation
Threshold	Heap size when next automatic garbage collection takes place. Gets recomputed after every garbage collection.
Size	Current heap size.
Active Size	Heap size after last garbage collection.
	Displays the three values from above.
Heap Parameters	
Field	Explanation
Minimal Size	Minimal heap size.
Free	Gives the percentage of free heap memory after garbage collection. For example, a value of 75% means that the heap threshold is set to approximately: $\text{Active Size} * 100 / (100 - 75)$ that is: $\text{Active Size} * 4$.
Tolerance	Gives the percentage by which the system is allowed for purposes of better memory allocation to increase the heap threshold.
Garbage Collector	
Field	Explanation
Active	Whether garbage collection is invoked automatically.

1.3 Information on Problem Solving

Finite Domain Constraints	
Field	Explanation
Variables Created	Number of finite domain variables created.
Propagators Created	Number of finite domain propagators created.
Propagators Invoked	Number of finite domain propagators invoked.
Spaces	
Field	Explanation
Created	Number of computation spaces created by <code>Space.new</code> .
Cloned	Number of computation spaces cloned by <code>Space.clone</code> .
Committed	Number of computation spaces committed by <code>Space.commit</code> .
Failed	Number of failed computation spaces.
Succeeded	Number of succeeded computation spaces.

1.4 Information on Programming Interface

Status Messages	
Field	Explanation
Idle	Whether messages are printed when the Mozart engine gets idle.
Garbage Collection	Whether messages on garbage collection are printed.
Output	
Field	Explanation
Maximal Depth	Maximal depth used for <code>System.show</code> and <code>System.print</code> (See also Chapter <i>Miscellaneous System Support: System, (System Modules)</i>).
Maximal Width	Maximal width used for <code>System.show</code> and <code>System.print</code> (See also Chapter <i>Miscellaneous System Support: System, (System Modules)</i>).
Errors	
Field	Explanation
Show Location	Whether error messages contain location information.
Show Hints	Whether error messages contain hints.
Maximal Depth	Maximal depth used for printing values in error messages.
Maximal Width	Maximal width used for printing values in error messages.
Maximal Tasks	Maximal number of tasks on the thread to be printed.

The Menu Bar

2.1 The **Panel** Menu

This menu contains operations to clear and close the panel and to stop the Oz system.

About...

Displays a window containing short information on the Panel.

Reset

C-r

Clears all monitoring information of the Panel.

Save Parameters...

Save the current system parameters settings to a file. Feeding that file resets the system parameters to their saved values.

Shutdown System...

After confirmation with a dialog the entire Oz system is halted.

Close

C-x

Closes the Panel window.

2.2 The **Options** Menu

Configure

Toggles whether the Panel provides for monitoring only, or for both monitoring and configuring.

Update...

Creates a dialog to set options when the Panel updates its display. **Update Time** gives the time period after which the display is updated. **Update Requirement** allows to toggle whether the display is updated, only if the mouse pointer is over the Panel's window.

History...

Creates a dialog to set the history range for the thread and memory graphs.

The `Panel` Object

This chapter lists all methods of the `Panel` object. New Panels can be created by creating new objects from the class `Panel.class`. Execution of

```
MyPanel = {New Panel.class init}
```

creates a new Panel which can be accessed by the variable `MyPanel`.

3.1 Invoking

open

```
{Panel.object open}
```

Invokes the Panel.

3.2 Options

option

```
{Panel.object option(configure +B)}
```

Toggles whether the panel allows to configure system parameters.

option

```
{Panel.object option(update time: +I<=_  
                        mouse: +B<=_)}
```

Sets the parameters for updating as described in Section 2.2.

option

```
{Panel.object option(history range: +I<=_)}
```

Sets the parameters for the history range as described in Section 2.2.

Index

Panel

`Panel, open, 7`

`Panel, option, 7`

Panel, 7

Property

`Property, get, 1`

`Property, put, 1`